Nichols Research Corporation
3452 Lake Lynda Drive, Suite 190
Orlando, FL 32817-1472
(407) 658-6550 FAX (407) 658-6681

Multisensor Fusion Using FLIR and LADAR Identification

Final Report

Alexander Akerman III and Ronald Patton
Nichols Research Corporation
3452 Lake Lynda Drive, Suite 190
Orlando, FL 32817
akermana@puzzler.nichols.com

Walter Delashmit
Lockheed Martin Vought Systems
P.O. Box 650003
Dallas, TX 75265-0003
delashmi@vs.lmco.com

Dr. Robert Hummel
New York University
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, NY 10012
hummel@relaxation.cs.nyu.edu

31 March 1997

U.S. Army Research Office

DAAH04-93-C-0049

Approved For Public Release;
Distribution Unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	31 March 1997	Final 29 Sep 93 - 31 Dec 97

4. TITLE AND SUBTITLE
Multisensor Fusion Using FLIR and LADAR Identification

5. FUNDING NUMBERS
DAAH04-93-C-0049

6. AUTHOR(S)
Alexander Akerman III & Ronald Patton
Walter Delashmit
Robert Hummel

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Nichols Research Corporation, 3452 Lake Lynda Dr., Suite 190, Orlando, FL 32817
Lockheed Martin Vought Systems, PO Box 650003, Dallas, TX 75265-0003
New York University, Courant Institute of Mathematical Sciences, 251 Mercer St., New York, NY 10012

8. PERFORMING ORGANIZATION REPORT NUMBER
N-TR-97-131

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 32364.1-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The primary objective of this project was to provide the Surrogate SemiAutonomous Vehicle (SSV) with a Demonstration II capability of performing automatic target recognition/identification (ATR/I). Detected objects of interest would be imaged with FLIR and/or LADAR sensors so the ATR/I algorithms must be compatible with either sensor, as well as exploit the synergy of processing both sensors simultaneously.

Our approach does not rely upon the precise coregistration of multiple sensors, but rather performs geometric hashing on the individual FLIR and LADAR images. Hashing represents an object by a collection of points, which are then matched to similarly constructed models. The matching is accomplished by iteratively selecting pairs of points, placing them in a Euclidean geometry coordinate system, concurrently translating and rotating all other object points to the same geometry, and then counting the number of occurrences of object and model points in the same cell. The geometric hashing software, originally developed for 2D SAR and FLIR imagery, has been extended to also accommodate 3D Ladar range and intensity imagery. The 2D hashing software was modified to allow up to ten dimensions. Currently, a 4D scheme is being used which represents (x,y) position, range, and intensity features.

14. SUBJECT TERMS		15. NUMBER IF PAGES
Geometric Hashing	Automatic Target Recognition	42
Sensor Fusion	Recognition-By-Components	16. PRICE CODE
FLIR	LADAR	

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

A Laplacian pyramid algorithm has been developed to fuse Ladar range and intensity imagery. Although previously used with dissimilar sensors (e.g., FLIR and TV), the algorithm proves maximally efficient with pixel coregistered imagery, as in the case of the two Ladar modes. Both target and background contrast and internal structure are significantly enhanced, thus facilitating target segmentation and feature extraction. The algorithm can be implemented with the existing pyramidal processors provided by Sarnoff Labs for RSTA image stabilization.

The FORTRAN/VAX software has been rewritten into C/UNIX and installed on a SUN SPARC processor within the Surrogate SemiAutonmous Vehicle (SSV). The C/UNIX software provides at least a 30-fold decrease in execution time and a 20-fold increase in model storage capacity.

Operational target recognition was performed during Demo C in July 1995 using both actual FLIR and synthetic LADAR imagery. 100% correct classification was obtained on the six target, 12 pose synthetic LADAR imagery which we had generated using the LARRA/SAIL and BRLCAD models. A subsequent laboratory experiment using the Demo C 66 FLIR target model set achieved an 86% correct classification of 56 unknown targets. Our software was installed on the Demonstration II SSV, which was exercised at Ft. Hood, Texas during the summer of 1996. Our algorithms/software provided the only RSTA automatic target recognition.

Our algorithm architectures fuses the results of the FLIR and LADAR hashing by using a Piecewise Level Fusion Classifier (PLFC). Target boundaries are extracted as an intermediate step in the determination of hash points (per edge curvature and edge intersection). Hence, we also perform multisensor fusion using combined FLIR and LADAR boundaries to perform Recognition-By-Components (RBC). We extend the work of others to provide viewpoint invariant recognition using perceptually organized features of geometric components. A Bayesian reasoning structure is used to fuse the results from the Hashing PLFC and the RBC algorithms.

We also determined an existing software suite for performing the Recognition-By-Components (RBC) algorithms. It is the Viewpoint Independent 3D Recognition and Extraction of Objects (VITREO) code developed at the U. of Central Florida. VITREO extracts geons from line images (including those created for hash point extraction), and then organizes the extracted geons into a database of recognized targets. VITREO is written in C and runs on a SUN UNIX platform

FOREWORD

This three year project was funded by DARPA/ISO under BAA93-01 for Autonomous Systems Technology as part of the Image Understanding (IU) program. The objective was to provide an unmanned ground vehicle with Reconnaissance, Surveillance, and Target Acquisition (RSTA) capabilities by onboard image processing of FLIR and LADAR imagery. The DARPA IU Program Manager was initially Dr. Oscar Firschein, and then Dr. Tom Strat for the last year. The ARO COTR was Dr. David Hislop.

The Unmanned Ground Vehicle (UGV) RSTA program provided the first opportunity to quantify geometric hashing performance in a military context, both with respect to various target types and on an operational platform, the Surrogate Semiautonomous Vehicle (SSV). Although several laboratory and field experiments have now been conducted, no overall assessment should be made until the results of an independent evaluation by the Army are completed. That evaluation is being conducted by the Night Vision and Electronic Sensors Directorate (NVESD). To date, the geometric hashing algorithm has exhibited very favorable performance and was the only recognition/identification software incorporated into the Demo II SSV's.

The 3-5μ FLIR produced imagery much different, and often of much lower quality, than that of 8-14μ FLIR's typically used for automatic target recognition (ATR). Hence, the data collections were insufficient for suitable model building. Too much emphasis was given to difficult conditions (e.g., vehicles on hills, obscuration, etc.) at the expense of not first generating a comprehensive target data base at precise orientations, ranges, and for a variety of times of day, year, and illumination conditions. The associated data basing was marginal, due to the ambitious data collection objectives and very limited resources made available to meet those objectives.

A shortcoming of the program was in not providing target imagery, either synthetic or real, to demonstrate a major RSTA requirement of target identification. That is, there were no foreign targets, even though operational scout personnel consistently stressed the need for a UGV to discriminate friendly from enemy vehicles.

The uncertainty of whether there would be a Ladar sensor onboard the UGV, much less what type of Ladar it would be, caused many discontinuities in developing the ATR algorithms, particularly those that involved the fusion of Ladar with other sensors. In the case of the geometric hashing algorithms, outstanding Ladar ATR performance was achieved against the synthetic LASER+ and LARRA/SAIL model-generated imagery. It is unfortunate that these models could not be tested operationally in the same manner as the FLIR ATR, i.e., on the Demo II SSV's.

The most important contribution of our team to the UGV RSTA program has been to transition the hashing software from a non-realtime, laboratory code to near real-time software resident on a SPARC workstation and thus operable in a military vehicle like the SSV. Notwithstanding any of the difficulties cited above, this ARPA initiative was an exciting and enjoyable experience that significantly pushed forward the ATR state of the art. We were very pleased to be a part of it.

Table of Contents

1. Statement of the Problem Studied 1

2. Summary of the Most Important Results 3

3. Ladar Intensity and Range Image Fusion 5

4. LADAR Hashing Models and Matching Results 8

4.1 LARRA/SAIL Synthetic LADAR Imagery 8

4.2 Hash Point Models 8

4.3 Test Results 10

4.4 2D Versus 3D Ladar Processing 10

4.5 2D Ladar Hashing of Three Actual Targets 12

5. SSV Sun SPARC Software 15

5.1 Code Conversion and Enhancement 15

5.2 Real Time Architecture 16

5.3 Efficient Hash Table Generation 16

5.4 Validation Examples 18

5.5 Performance Metrics 18

6. FLIR Hashing Models and Matching Results 19

6.1 FLIR Imagery Used for Model Building 19

6.2 Image Enhancement Algorithms 22

6.3 Hash Point Models 22

6.4 FLIR Hashing Test Results 24

7. List of Publications and Technical Reports 32

8. List of Presentations and Demonstrations 33

9. List of Participating Scientific Personnel 34

10. References 35

Appendix A. Geometric Hashing Evolution 36

Appendix B. Theoretical Formulation for Hashing of Ladar Imagery 39

List of Tables

Table 1. Matching Criteria for Hashing MLRS Live/M2 Model 10
Table 2. 2D Hashing Results for MLRS Live/M2 Models 12
Table 3. Hashing Point Corresponding Between MLRS Live and M2 Model 13
Table 4. 2D Hashing Recognition Results for Tower Test Ladar Imagery (P_{CC} = 73.4%) 14
Table 5. First RSTA FLIR Target Training Set - 6 & 7 October 1994 20
Table 6. Second RSTA FLIR Target Training Set - 7 October 1994 21
Table 7. Subjective Evaluation of Training Image Quality 21
Table 8. Overall Classification Matrix 29
Table 9. Overall Classification Probability Matrix (P_{cc} = 86%) 29
Table 10. Classification Results for M35 Truck Target (12 of 12) 29
Table 11. Classification Results for HMMWV Target (8 of 12) 30
Table 12. Classification Results fro M113 APC Target (9 of 12) 30
Table 13. Classification Results for M543 Wrecker Target (10 of 10) 31
Table 14. Classification Results for M60 Tank Target (9 of 10) 31

List of Figures

Figure 1. Overall Architecture for Multisensor Fusion Using FLIR and LADAR Identification 2
Figure 2. Ladar Image Enhancement By Image Fusion 6
Figure 3. Additional Examples of Ladar Image Enhancement 6
Figure 4. Baseline Feature Algorithms Applied to Ladar Imagery 7
Figure 5. LADAR Line Segments and Extracted Hash Points for M113 APC's 9
Figure 6. LADAR Line Segments and Extracted Hash Points for M60 Tanks 9
Figure 7. Synthetic Ladar Image of an MLRS and Corresponding Hash Point Model 11
Figure 8. Synthetic Ladar Image of an M2 IFV and Corresponding Hash Points 11
Figure 9. FLIR Image Enhancement and Geometric Hashing of an M35 Truck 23
Figure 10. HMMWV Images and Point Models 25
Figure 11. M113 APC Images and Point Models 26
Figure 12. M35 Truck Images and Point Models 26
Figure 13. M543 Wrecker Images and Point Models 27
Figure 14. M60 Tank Images and Point Models 27

1. Statement of the Problem Studied

The primary objective of this project was to provide the Surrogate SemiAutonomous Vehicle (SSV) with a Demonstration II capability of performing automatic target recognition/identification (ATR/I). Detected objects of interest would be imaged with FLIR and/or LADAR sensors so the ATR/I algorithms must be compatible with either sensor, as well as exploit the synergy of processing both sensors simultaneously. Our approach does not rely upon the precise coregistration of multiple sensors, but rather performs geometric hashing on the individual FLIR and LADAR images.

Geometric hashing is the fundamental technique which we have applied to the military ATR problem [Akerman, et al., 1992]. Conceived by researchers at the NYU Courant Institute [Lamdan and Wolfson, 1988], hashing represents an object by a collection of points, which are then matched to similarly constructed models. The matching is accomplished by iteratively selecting pairs of points, placing them in a Euclidean geometry coordinate system, concurrently translating and rotating all other object points to the same geometry, and then counting the number of occurrences of object and model points in the same cell.

Geometric hashing is particularly appealing since it can be very efficiently implemented with parallel processing. An unknown object can be simultaneously tested against thousands of models, including specific orientations/states of each target [Bourdon and Medioni, 1990].

Our contribution to hashing algorithms has been the application of the technique to military targets in Synthetic Aperture Radar (SAR), Forward Looking Infrared (FLIR) imagery, and LADAR imagery. During 1988-1990, we developed a SAR point extractor, determined thresholds and tolerances for SAR point matching, and created software for simultaneous multiple model testing [Akerman and Patton, 1990].

In 1991, we began an investigation of FLIR imagery hashing, with particular emphasis on the extraction of robust hash points from the targets. Algorithms were developed to select points that represented the target's geometrical structure and that were thus stable and repeatable under various radiometric conditions, due both to the external environment and to the target itself. These algorithms entailed first extracting significant contours corresponding to the target's key components (tread, turret, etc.). The hash points are obtained from the end points, intersections, and key curvature of those lines.

In 1993, we refined the FLIR-associated hashing algorithms under a contract with the Army Night Vision and Electronic Systems Directorate [Akerman et. al, 1993]. Specifically, the algorithms were extended to second generation FLIR imagery which provides much greater detail of target internal structure.

As shown by Figure 1, our algorithm architectures fuses the results of the FLIR and LADAR hashing by using a Piecewise Level Fusion Classifier (PLFC). Such fusion is based upon the work of Thomopoulos, [1987].

Target boundaries are extracted as an intermediate step in the determination of hash points (per edge curvature and edge intersection). Hence, we also perform multisensor fusion using combined FLIR and LADAR boundaries to perform Recognition-By-Components (RBC). We extend the work of Lowe [1985], Biederman [1987], and others to provide viewpoint invariant recognition using perceptually organized features of geometric components. A Bayesian reasoning structure is used to fuse the results from the Hashing PLFC and the RBC algorithms.

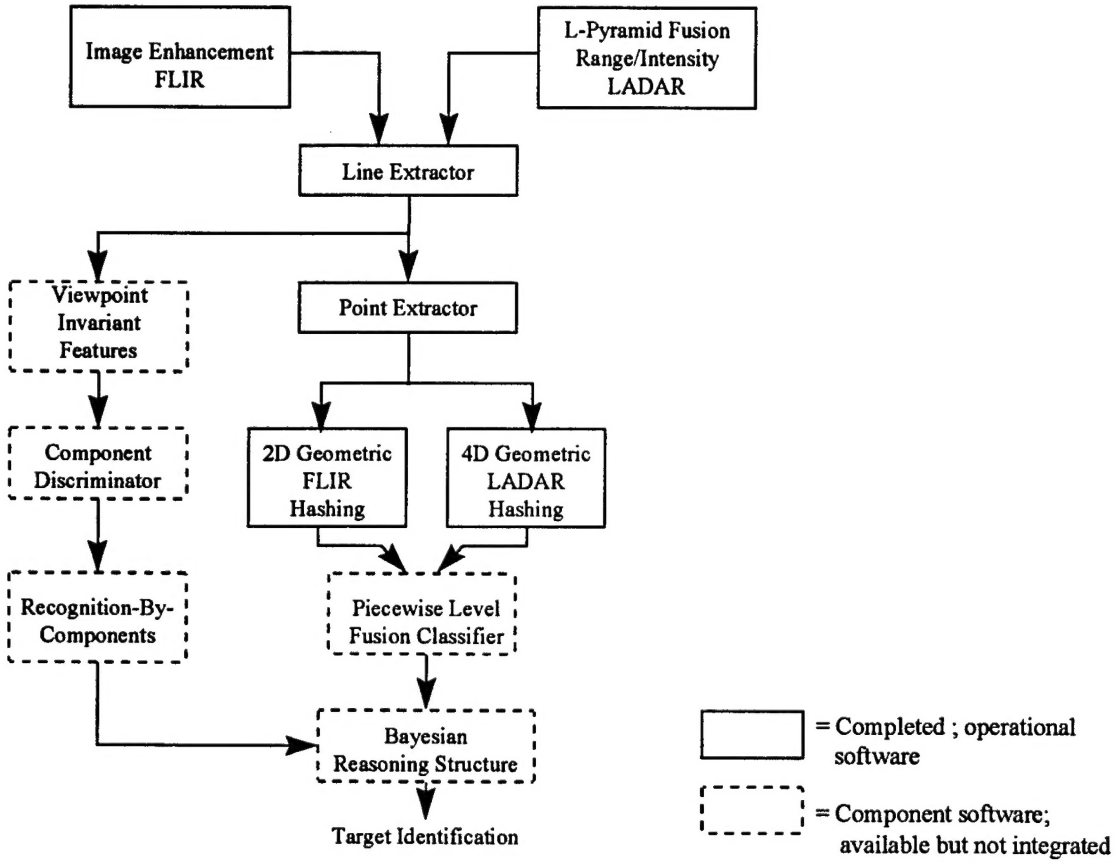


Figure 1. Overall Architecture for Multisensor Fusion Using FLIR and LADAR Identification

2. Summary of the Most Important Results

As indicated, completed work is shown by the solid blocks in Figure 1. The algorithms were not finished because $100K (of $300K total) was deleted from our final year funding. As the prime contractor, Nichols Research Corporation (NRC) developed the overall architecture and the front end FLIR and LADAR image enhancement algorithms, as well as the geometric hashing codes. Our techniques for LADAR image enhancement fuses the pixel coregistered range and intensity images by merging the individual levels of a Laplacian Pyramid decomposition of each of the LADAR images. The FLIR image enhancement uses classical histogram equalization nonlinear mapping, and gradient sharpening techniques.

The geometric hashing software, originally developed for 2D SAR and FLIR imagery, has been extended to also accommodate 3D Ladar range and intensity imagery. The 2D hashing software was modified to allow up to ten dimensions. Currently, a 4D scheme is being used which represents (x,y) position, range, and intensity features.

A Laplacian pyramid algorithm has been developed to fuse Ladar range and intensity imagery. Although previously used with dissimilar sensors (e.g., FLIR and TV), the algorithm proves maximally efficient with pixel coregistered imagery, as in the case of the two Ladar modes. Both target and background contrast and internal structure are significantly enhanced, thus facilitating target segmentation and feature extraction. The algorithm can be implemented with the existing pyramidal processors provided by Sarnoff Labs for RSTA image stabilization.

The FORTRAN/VAX software has been rewritten into C/UNIX and installed on a SUN SPARC processor within the Surrogate SemiAutonmous Vehicle (SSV). The C/UNIX software provides at least a 30-fold decrease in execution time and a 20-fold increase in model storage capacity.

Operational target recognition was performed during Demo C in July 1995 using both actual FLIR and synthetic LADAR imagery. 100% correct classification was obtained on the six target, 12 pose synthetic LADAR imagery which we had generated using the LARRA/SAIL and BRLCAD models. A subsequent laboratory experiment using the Demo C 66 FLIR target model set achieved an 86% correct classification of 56 unknown targets.

Our software was installed on the Demonstration II SSV, which was exercised at Ft. Hood, Texas during the summer of 1996. Our algorithms/software provided the only RSTA automatic target recognition. As an adjunct to this activity, the FLIR target model set was expanded to include 12 poses (every 30 degrees azimuth) of the M1 tank.

Subsequent to Demo II, our algorithms were selected for independent evaluation by the Army Night Vision and Electronic Systems Directorate (NVESD). Again, only our algorithms were chosen for the RSTA ATR evaluation. To accommodate this assessment, we had to completely reengineer the software from its SSV-based, Khoros 1 configuration to be compatible with the NVESD laboratory computer system. This entailed an exhaustive effort to accommodate run scenario and image interfaces (encompassing five different camera configurations) as well as additional setup command software to allow batch processing.

We also determined an existing software suite for performing the Recognition-By-Components (RBC) algorithms. It is the Viewpoint Independent 3D Recognition and Extraction of Objects (VITREO) code developed at the U. of Central Florida. VITREO extracts geons from line images (including those created for hash point extraction), and then organizes the extracted geons into a database of recognized targets. VITREO is written in C and runs on a SUN UNIX platform

3. Ladar Intensity and Range Image Fusion

We have previously developed an algorithm for fusing FLIR and TV, (as well as laser intensity and TV) images into a single fused image [Akerman, 1992]. In essence, the algorithm represents each image as a Laplacian pyramid [Burt and Adelson, 1983], and then combines the individual sensor representations one level at a time using an appropriate pixel select criteria [Toet et al, 1989].

The resultant image quality is significantly dependent upon pixel coregistration between the two individual images. Hence, one might expect optimal results in the case of Ladar range and intensity imagery from the same sensor and thus exactly pixel coregistered. Figures 2 and 3 indeed illustrate that such an image fusion does provide a significant enhancement in target and other scene detail.

The upper subimage is the Ladar intensity return. Notwithstanding numerous pixel "dropouts," this image provides good internal detail of objects. However, those objects often lack distinct borders and instead blend with their immediate background. In particular, note the wheeled object in the lower left quadrant of Figure 2. It could be very difficult for an automatic target recognizer to discern that it is a truck.

The middle subimage is a transform of the Ladar range image, in which a zero gray scale (Black) is the ground plane and higher Gray Scale values represent increasing height above the ground plane. For this representation, internal object structure is minimized, particularly near the ground plane. However, the overall shape silhouette is enhanced.

The bottom subimage portrays the resultant image fusion. The wheeled object in Figure 2 is now clearly a truck. The other vehicles in that image are also more distinct. Figure 3 presents two other scenes. For the one on the left, the target was already very distinctive in the intensity image, so the merging provides no significant enhancement. In comparison, however, note the detail in the foliage of the merged image as compared to that in either of the intensity or elevation images.

For the scene on the right side of Figure 3, note the truck next to the tree. The truck is not very distinct in either the intensity or the elevation images, but is "pulled out of the mud" in the merged image.

In all instances that we have thus far processed, the merged image never has an object of lower image quality than that of the intensity or elevation image alone. Often, there is a very significant improvement as we have shown. Hence, all of our geometric hashing algorithms are being applied only to Ladar imagery that has merged both the intensity and range signatures.

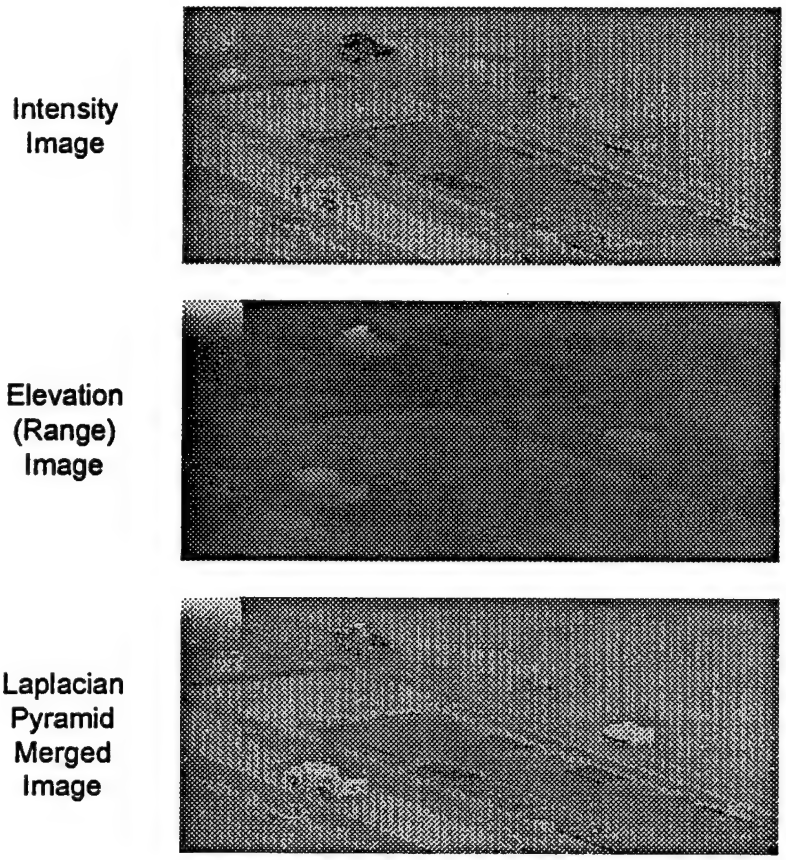


Figure 2. Ladar Image Enhancement By Image Fusion

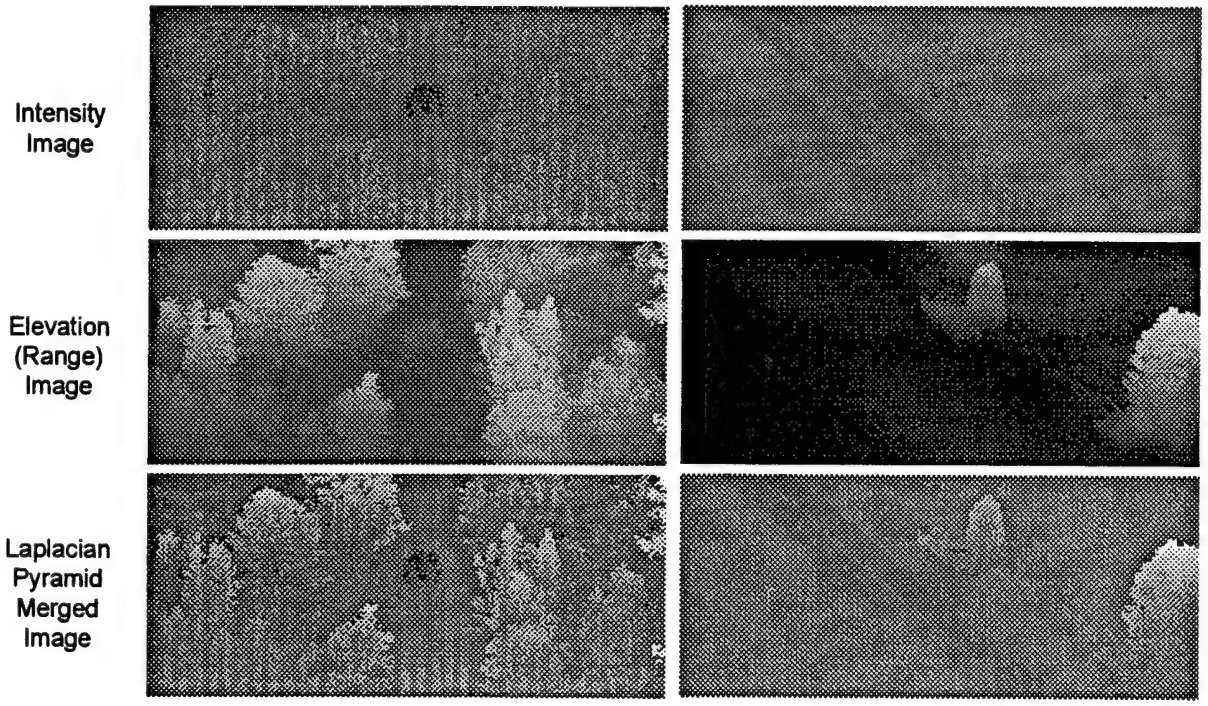


Figure 3. Additional Examples of Ladar Image Enhancement

Figure 4 illustrates the application of the Rule-Based Line and Point Extraction algorithms applied to Ladar imagery of four target types. This imagery was collected with a Loral Vought diode pumped laser radar operating at 1.06μ, with a 0.4mr horizontal and vertical angular resolution, and with a 0.15m range resolution. The targets are at ranges of 300-400m and at depression angles of 14-18°.

The line extraction algorithms were applied to both the Laplacian Pyramid fused image (of the Ladar intensity and elevation images) and to the elevation image alone. Note that while the Laplacian fused image yields significant internal detail, it does not capture all of the target's exterior boundary. (Although not shown, this deficiency is significantly worse when only the Ladar intensity image is used). Conversely, the Ladar elevation image yields a good segmentation of the overall target shape but loses much of the internal target detail.

When both line extractions are combined together, all of the key geometrical components of the target are distinctly outlined. Note also that there are no extraneous lines on the target, except when there are obscuring clutter artifacts. Hence, the line segmentation provides a very robust geometry for the extraction of the hash points, which are also shown in Figure 4.

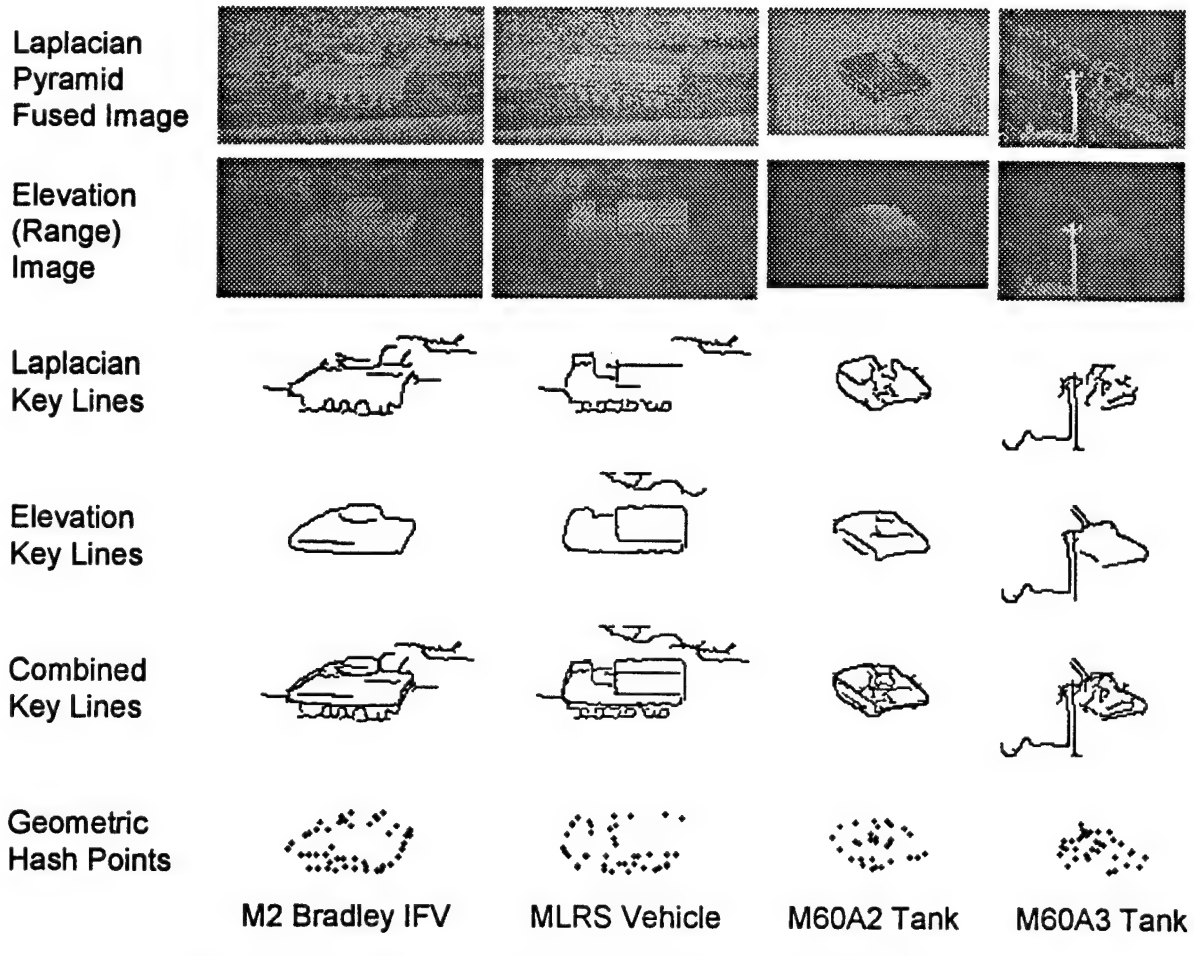


Figure 4. Baseline Feature Algorithms Applied to Ladar Imagery

4. LADAR Hashing Models and Matching Results

4.1 LARRA/SAIL Synthetic LADAR Imagery

Synthetic LADAR range images were generated by NYU from the Ballistic Research Laboratory (BRL) computer aided design (CAD) models for six tactical vehicles (targets) using the Laser Radar Recognition Algorithm (LARRA)/Synthetic Assembly Image Layout (SAIL) modeling code. The targets selected were the M60A3 tank, M113 armored personnel carrier (APC), M35 truck with a rack or a canvas cover, M35 truck without a rack or canvas cover, HMMWV troop version with the conventional sloped rear and HMMWV cargo version which has a square back. The rationale used for selection was the availability of the appropriate BRLCAD models for unrestricted access. These images were generated at a sensor depression angle of zero degrees to correspond to the Unmanned Ground Vehicle (UGV) scenario. Images were created every 15 degrees from zero to 360 degrees. The original images were created using a resolution of 0.05 milliradians (mr) in both azimuth and elevation with each pixel corresponding to a ray trace. All output images are in the Khoros viff format. For these six targets, images at every 30° were selected to use to build up a data base of 72 models consisting of 12 orientations for the 6 targets.

4.2 Hash Point Models

Image chips, line extraction/segmentation and extracted hash points are shown in Figure 5 for three different orientations of the M113 APC corresponding to aspect angles of 285°, 15°, and 270°. The white pixels in the image chips correspond to range data drop outs where a very large nontarget value was recorded. It should be noted that the line and point segmentation works even for cases where a complete target outline in terms of edge/line structure is not obtained. It can also be clearly seen in this figure that line end points, line intersections and points of curvature have all been extracted for the models.

Similar results are also shown in Figure 6 for three different orientations of the M60 tank corresponding to aspect angles of 150°, 90°, and 75°. For this case, the tank outline is less complete than for the M113 due to much missing structure along the bottom of the tank tread. In addition, one background point was picked up for the M60 tank at 90° aspect, but this single extraneous point will have little or no effect on match results.

Visual comparison of Figures 5 and 6 clearly shows dissimilarities in the extracted point structure for the two different targets (M113 and M60). This dissimilarity in the extracted features, i. e., hash points, is the basis on which the geometric hashing algorithms are used for target identification.

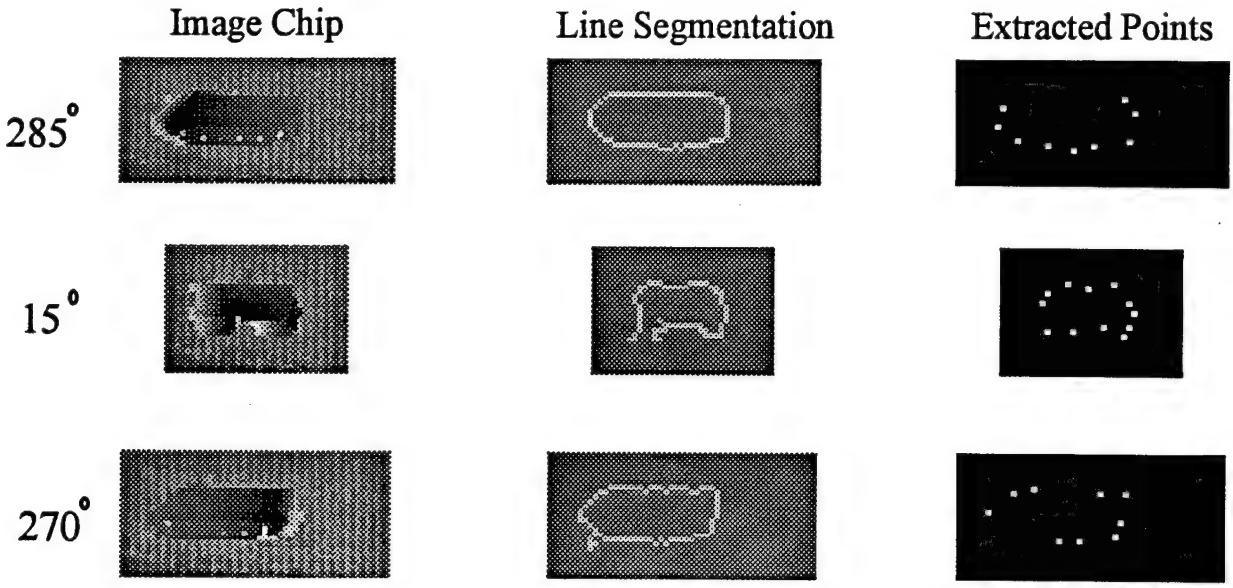


Figure 5. LADAR Line Segments and Extracted Hash Points for M113 APC's

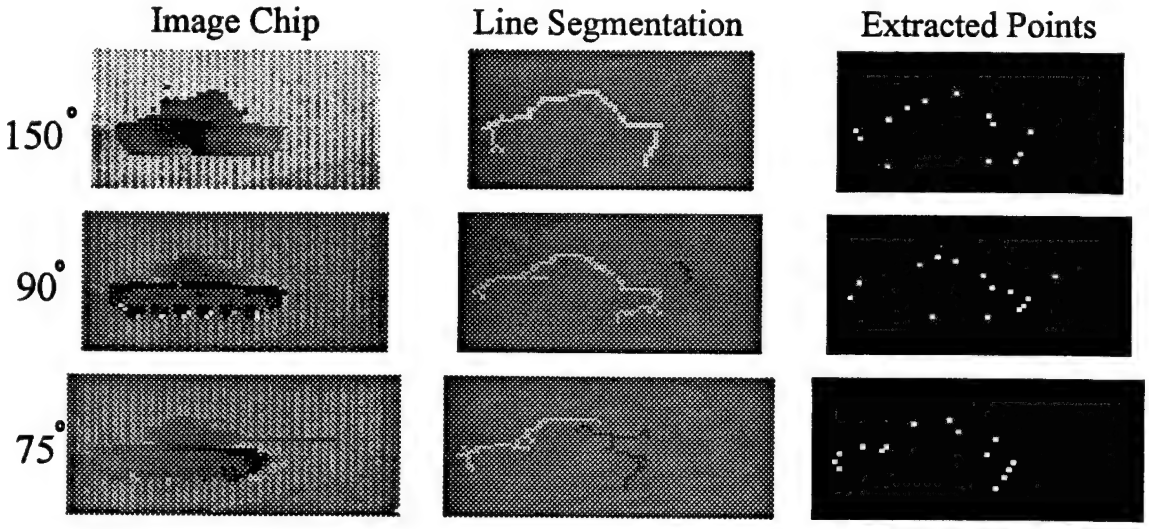


Figure 6. LADAR Line Segments and Extracted Hash Points for M60 Tanks

4.3 Test Results

The geometric hashing algorithms were tested in a real-time operational scenario during Demo C on July 27, 1995. Due to operational time constraints for the entire demonstration of our work and the work of other contractors, only five LADAR images were processed. The five images corresponded to a M60 tank at 270° aspect, a M113 APC at 15° aspect, a HMMWV cargo version at 15° aspect, a M35 truck with the rack and canvas top at 300° aspect, and another M60 tank but with this tank at 120° aspect. The results of this real-time operational test was that the geometric hashing algorithms produced 100% correct target recognition. It should be noted that this realtime testing was a recognition task rather than an identification task since the available target set did not support identification, i. e., there were not multiple types of a single class of targets (e.g., M60, M1, T62, T72, tanks, etc.).

4.4 2D Versus 3D Ladar Processing

To perform this analysis, we used synthetic imagery generated by our LASER+ model. We selected an MLRS (Multiple Launch Rocket System) vehicle at 1500m as the live image, and hashed it against an M2 Bradley Infantry Fighting Vehicle model, also at 1500m. Figures 7 and 8 show a 2D projection of those two targets, along with the corresponding hash points that were extracted. First, we hash only in a 2D space to see if a match would occur. Per Table 1, we allowed the percentage of match threshold to be set at 40%, the point match distance threshold to be no more than 1.5 pixels, and the orientation (image rotation) mismatch to be within ten degrees.

Table 1. Matching Criteria for Hashing MLRS Live/M2 Model

Percentage of Match Threshold	= 40%	Intensity Thresh Used (If Valid)	= 0
Point Match Distance Threshold	= 1.50	Use Range Value (1=Yes, 0=No)	= 0
Point Match Tolerance Allowed+/-	= 1	Range Threshold Used (If Valid)	= 0.0
Orientation Mismatch Threshold	= 10 Deg	% of Live Points To Use As Masters	=100%
Stop On 1st Match (1=Yes, 0=No)	= 0	# Of Live Points To Use As Masters	= 20
Apply Affine Trans. (1=Yes, 0=No)	= 0	% Of Live Points To Use As Slaves	=100%
Use Intensity Value (1=Yes, 0=No)	= 0	# Of Live Points To Use As Slaves	= 20

The (master, slave) = (11,4) live hash point set (listed on the last line of Table 2) is chosen because the percent match is the foremost criteria, provided that the average point distance and the orientation delta thresholds are also met. In this case, the values are 1.17 pixels and 4 degrees, both of which are below threshold and thus are acceptable. Note that (master, slave) = (19,10) live hash point set has a much lower average point distance (0.52 pixels) and orientation delta (0°), but was not chosen because the percent of matched live points (55%) was less than the (11,4) master, slave set.

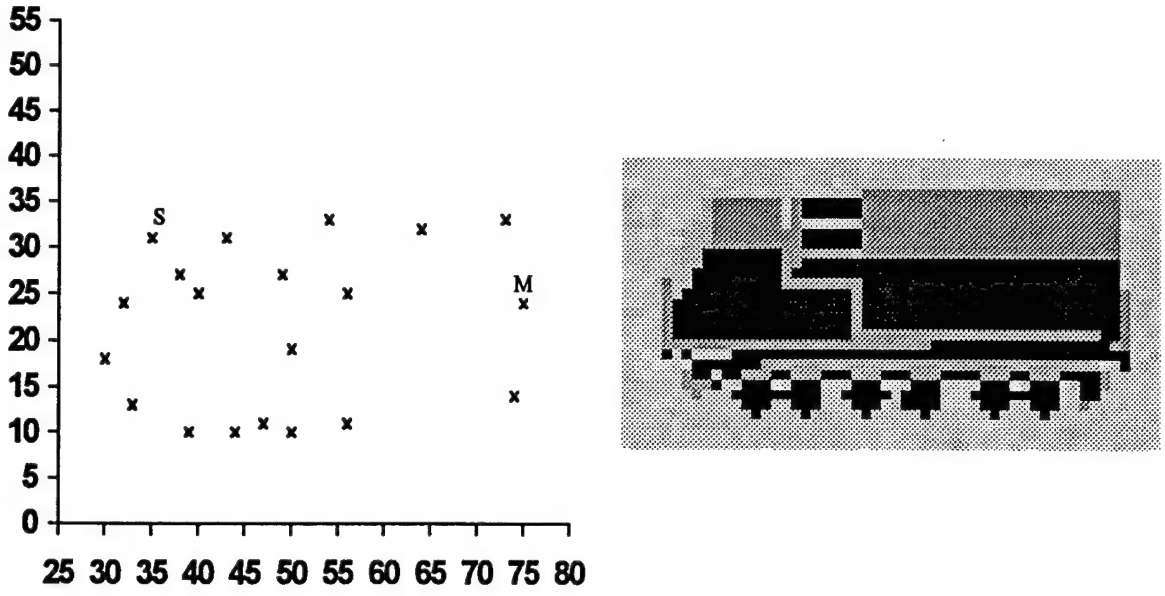


Figure 7. Synthetic Ladar Image of an MLRS and Corresponding Hash Point Model

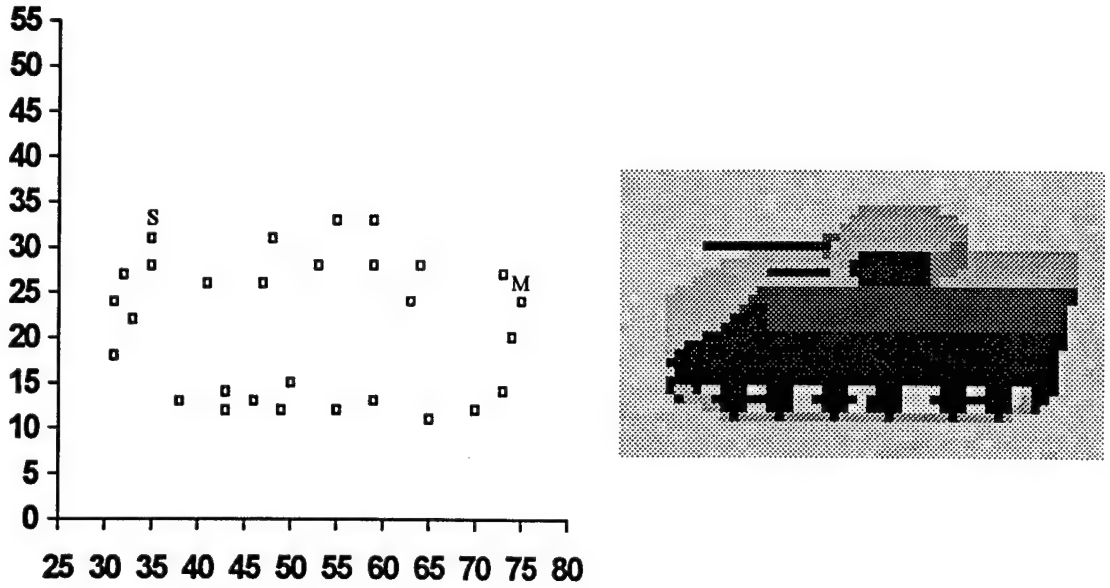


Figure 8. Synthetic Ladar Image of an M2 IFV and Corresponding Hash Points

Table 2. 2D Hashing Results for MLRS Live/M2 Models

LIVE/MODEL SELECTION SUMMARY

LIVE		% OF		MODEL		% OF	POINTS	AVG POINT	ORIENTATION	
MASTER	SLAVE	MATCH	No.	MASTER	SLAVE	MATCH	MATCHED	DISTANCE	MODEL	LIVE
13	14	45%	1	15	23	31%	9	0.9785	180	187
13	7	45%	1	15	12	31%	9	0.8750	180	189
13	9	45%	1	15	10	31%	9	0.8017	180	190
13	9	45%	1	15	14	31%	9	0.7285	180	190
13	2	50%	1	17	5	34%	10	1.0729	180	181
13	2	50%	1	17	7	34%	10	0.9618	180	179
16	19	50%	1	24	26	34%	10	0.6476	180	180
16	10	50%	1	24	11	34%	10	0.6015	180	180
19	10	55%	1	26	11	38%	11	0.5242	180	180
11	4	60%	1	10	4	41%	12	1.1726	180	176

Table 3 summarizes the point-to-point matching of the 29 points of the MLRS model by the 20 points of the M2 live "unknown". Any row entry of Table 3 that labels with "-1" the live point number (and all successive entries) means that that corresponding model point was not matched. **The results of this 2D hashing is that the live MLRS would have been identified as an M2 for this experiment, because more than 40% of the model point were matched.**

The erroneous identification result is prevented when the hashing is extended into 3D, making use of the range associated with each point. No such error occurs when the range difference threshold is set at 0.15 meters. (That particular value was chosen since it corresponds to the range resolution of the Ladar). Referring to Table 3, note that only 7 of the points matched, neither of which exceeds the 40% point match criteria.

4.5 2D Ladar Hashing of Three Actual Targets

The previous subsection illustrated 2D and 3D hashing for the MLRS and M2 targets using synthetic imagery created by our LASER+ computer model. We also performed similar 2D hashing on Loral Vought actual Ladar imagery of three target types: an M113 armored personnel carrier (APC) and two very similar tanks (M60A2 and M60A3). Examples of the tank imagery and corresponding hash points are shown by Figure 4. (Also shown in this figure are the two synthetic LASER+ targets, the M2 Bradley and the MLRS, which were used in the previously described 3D hashing test. In Figure 4, these synthetic targets have been inserted into the same type images as those of the M60's). This hashing experiment used five M113 and fourteen M60 images. All of the M60's were at a nominal range of 300m, yielding approximately 35 hash points for each of those targets. However, the M113's, which are smaller in size than the M60's, were all at a longer range of 400 meters. Those conditions yielded an average of only ten hash points per M113, which is the significant factor causing the hashing misidentifications. That is, all target type mismatches are M113 associated.

As shown by Table 4, six models were built from the 19 images. Those six models consist of one M113, and five M60's each at a different azimuth. All 19 images (including the six used for model creation) were hashed against each of the six models. The match criteria were (1) at least 50% correspondence of the live or model points, (2) less than an average of 1.5 pixels for the separation of the matched points, and (3) less than 20° orientation error for rotations within the image planes of the aligned live and model points.

Table 3. Hashing Point Corresponding Between MLRS Live and M2 Model

MODEL POINT	MODEL ROW	MODEL COL	MODEL INTENSITY	MODEL RANGE	LIVE POINT	LIVE ROW	LIVE COL	LIVE INTENSITY	LIVE RANGE	POINT DISTANCE	INTENSITY DIFF	RANGE DIFF
1	22	55	87	1499.8	1	22	54	87	1500.3	1.00	0	0.5
2	22	59	87	1499.8	-1	-1	-1	-1	-1	-1.00	-1	-1.0
3	24	48	54	1499.0	-1	-1	-1	-1	-1	-1.00	-1	-1.0
4 S	24	35	253	1499.2	4 S	24	35	87	1499.5	0.00	166	0.3
5	27	64	137	1498.8	-1	-1	-1	-1	-1	-1.00	-1	-1.0
6	27	53	240	1498.1	-1	-41	-1	-1	-1	-1.00	-1	-1.0
7	27	59	240	1498.1	-1	-1	-1	-1	-1	-1.00	-1	-1.0
8	28	73	87	1499.5	-1	-1	-1	-1	-1	-1.00	-1	-1.0
9	27	35	73	1500.4	-1	-1	-1	-1	-1	-1.00	-1	-1.0
10 M	31	75	226	1498.4	11 M	31	75	240	1497.9	0.00	14	0.5
11	28	32	73	1500.1	-1	-1	-1	-1	-1	-1.00	-1	-1.0
12	29	41	226	1498.3	8	30	40	252	1498.1	1.41	26	0.2
13	29	47	226	1498.3	-1	-1	-1	-1	-1	-1.00	-1	-1.0
14	31	63	226	1498.3	-1	-1	-1	-1	-1	-1.00	-1	-1.0
15	31	31	73	1499.1	10	31	32	252	1498.1	1.00	179	1.1
16	35	74	226	1498.1	-1	-1	-1	-1	-1	-1.00	-1	-1.0
17	33	33	240	1498.1	-1	-1	-1	-1	-1	-1.00	-1	-1.0
18	37	31	240	1498.1	13	37	30	87	1498.2	1.00	153	0.1
19	41	73	240	1498.2	14	41	74	240	1498.3	1.00	0	0.1
20	40	50	240	1498.2	-1	-1	-1	-1	-1	-1.00	-1	-1.0
21	41	43	240	1498.5	-1	-1	-1	-1	-1	-1.00	-1	-1.0
22	42	59	240	1498.6	-1	-1	-1	-1	-1	-1.00	-1	-1.0
23	43	70	240	1498.5	-1	-1	-1	-1	-1	-1.00	-1	-1.0
24	42	46	240	1498.6	16	44	47	240	1498.6	2.24	0	0.0
25	42	38	240	1498.7	18	45	39	240	1498.7	3.16	0	0.0
26	43	43	240	1498.7	19	45	44	240	1498.7	2.24	0	0.0
27	43	49	240	1498.7	20	45	50	240	1498.7	2.24	0	0.0
28	43	55	240	1498.7	17	44	56	240	1498.6	1.41	0	0.0
29	44	65	240	1498.7	-1	-1	-1	-1	-1	-1.00	-1	-1.0

Table 4. 2D Hashing Recognition Results for Tower Test Ladar Imagery (P_{CC} = 73.4%)

		MODELS						
		M113 120°	M60A2 310°	M60A3 120°	M60A3 230°	M60A3 270°	M60A3 330°	NO MATCH
UNKNOWNS								
M113	120°	3			1	1		
M60A2	310°	1	1					
M60A3	120°			2				1
M60A3	230°				2			
M60A3	260°					2		
M60A3	270°					3		
M60A3	330°	1					1	

Except for M60A3 at 260°, each unknown set included the model.

Note that although the targets are at different ranges, all the hashing operations scaled the point sets to a common 300m range. Unlike the synthetic imagery (of the M2 and MLRS targets) used in the previous subsection, however, the range scaling does not assure that the M113 and M60 targets are positioned precisely at the same point. As such, differences in the range (z) values of the (x,y) hash points could not be used directly to disqualify a point match. Rather, an additional algorithm is needed to compare the relative range differences between the model and the live (unknown) points so as to allow 3D hashing. That algorithm was under development but could not be completed within the reduced funding allocated to this contract.

5. SSV Sun SPARC Software

Existing FORTRAN geometric hashing code was converted to C and enhanced to run in real time on the Sun SPARC computers which are available on the SSVs. This C UNIX software was delivered to Lockheed-Martin and support was also provided to install and checkout the software to ensure proper operation of the code. This effort was crucial to the successful demonstration for Demo C as discussed above.

5.1 Code Conversion and Enhancement

The original FORTRAN code was converted to C to run under an UNIX Sun workstation. This was in accordance with the contract requirements for the Autonomous Systems Technology (AST) Program. A LINUX version (PC UNIX) is also available. Various enhancements and updates to the code were made during the conversion process. Some of these changes were made to make the code more efficient for real time implementation, some were made as a result of SSV requirements specified in the Lockheed-Martin Software Configuration Control Document [Severson, 1995], some were made to make the code more portable and essentially machine independent, some were made to increase the numerical precision of the code, and some were made to update the original version of the code. This code development refers not only to the code hosted on the SSV for real time usage, but also to code which was developed to produce the hashing model files. Structured C programming techniques were used with extensive documentation embedded in the actual code.

To enhance real time operation, the model file is read in from memory at the start of the program by the executive controller and maintained in memory during system operation. This results in significant speed enhancement since all software and data are memory resident. This software is configured so that the model file can be stored and read as either a binary file or an ASCII file. It is preferable to use binary files which results in reduced storage requirements and enhanced speed and is compatible with the requirements specified by the Software Configuration Control Document and the intended structure and use of the executive controller.

Code updates were made in regard to what version of the feature extraction parameters were used to ensure total program consistency for all data. In addition, some multiple rotations were combined into a single equivalent rotation. In addition, judicious use of either in line code or function calls were embedded into the code structure. These techniques enhanced both speed of computation and numerical accuracy. A range scaling algorithm was also developed to ensure that no range mismatch existed between the prestored model data base and the operational imagery.

The geometric hashing code is used to perform target recognition or identification based on using a detected potential target and performing processing over an image chip. The present size of the image chip is restricted to be 180 pixels by 180 pixels, or actually the product of the rows and columns for the image chip must be no greater than 180^2 or 32,400. There is no requirement to have a square image chip and in fact image chips are usually rectangular as used for the actual demonstration. The true limitation at present is that the actual product of the rows times the columns be restricted to a value of 32767. This is only because certain variables in the program have been declared as type "short" rather than type "int." This is not considered a

program limitation since the specified size should be adequate for standard operational modes. It would also be easy to change these variable types should larger size image chips be designated for processing.

5.2 Real Time Architecture

The real time architecture was designed to be compatible with the Lockheed-Martin Software Configuration Control Document. As such all input and output is performed in the main program body and the geometric hashing code is called as a function from this main program. This main program provides the hashing points from the feature extraction software which have been hosted on either the SSV Sun SPARC computer or on the SSV Distributed Array Processor (DAP). The model file is also read in by this main program. The output results are transferred back to the main program which then outputs the classification results for target class and confidence measure and the image chip along with the line segmentation and point images for display purposes. No calls to "exit" are made internally to any of the hashing routines.

5.3 Efficient Hash Table Generation

The hash table and model file generation is all performed internally to memory for all operations. No specialized commercial software is used for any of these functions. A previous version of the code used the CINDEX software developed by Trio Systems Inc. This was required in the previous system based on the hosting processor. However, this specialized software is somewhat machine dependent and would have to be licensed for any machine on which the software were to be hosted. These complications were avoided by using a structured key model technique with all computations being performed internally.

The previous FORTRAN keyed-access file used to store the patterns associated with each master-slave point pair was composed of an (X,Y) coordinate that uniquely specified a record in the file. Interactive disk accesses are not appropriate for real time operation and would be inconsistent with the software configuration requirements. Thus this process was replaced with a memory resident equivalent while maintaining a similar interface for access to individual records.

There are four record types in a hash file corresponding to the following:

- the master record,
- the descriptor records,
- the feature records, and
- the coordinate records.

There is only one master record and thus no key is required for its access. A descriptor file and a feature record exist for each target that is represented in the hash table so they can be represented with an integer that corresponds to a particular target number in the model data base. The coordinate records comprise the majority of the hash table and their storage and access requires special attention. Conceptually, the key can still be considered to be composed of an (X,Y) coordinate pair. If m, d_x, f_x, and c_x are addresses of memory locations for the respective record types, the layout in memory of a hash table can be illustrated as follows:

m → master record

d_1 → target 1 descriptor record
d_2 → target 2 descriptor record
d_3 → target 3 descriptor record
.
.
.
d_N → target N descriptor record

f_1 → target 1 feature record
f_2 → target 2 feature record
f_3 → target 3 feature record
.
.
.
f_N → target N feature record

c_1 → (0,1) coordinate record
c_2 → (0,3) coordinate record
c_3 → (0,1) coordinate record
.
.
.

This shows the techniques for the memory resident storage and access which was a key to the real time code implementation. The particular details of each record at this stage are not important, but they can be ascertained by looking at the internal code structure.

The software supports both the usage of ASCII or binary model files. However, it is recommended that binary files be used since this offers a factor of 3 to 5 reduction in storage size over ASCII files and results in a speed decrease during operation. Since problems often arise when transferring binary data files to different machines, the initial model files are developed and stored as ASCII files for transferring and executable source code is sent with the files to convert them to binary on the hosting machine. This procedure was used very successfully for this effort and has significantly alleviated some earlier problems in the development process.

Another very important aspect of this code is that the same software used in real time operation is used to develop new hash models. This allows new unknown models to be rapidly added to the data base even under field operation conditions. A soldier could find a new target type, collect sufficient image data to convince him that he had a good representation of the model in terms of hash points and then add this to the model data base as a new target type. Of course, the proper designation of the target type and format would be needed, but these items can be added by an user with modest training.

5.4 Validation Examples

During and after code development, three test cases corresponding to previous control cases used by Nichols Research Corporation (NRC) for their initial FORTRAN version of the code were used to verify code operation. These cases corresponded to two M60 tanks at different aspects and one M113 APC. All results agreed within computational precision after updating both code versions with agreed upon changes. The computational precision was based on differences in using PCs and Sun workstations at Loral Vought and initially using a MicroVAX II at Nichols Research Corporation for these previous cases. The previously discussed combining of multiple rotations into a single rotation also resulted in some small differences.

After developing the code, extracting the appropriate hashing features, and developing the model file, a complete end to end run was made on 22 cases at Loral Vought before the code was installed at Lockheed-Martin on the SSV computers. After installation and set up on the SSV computers, three different test cases were run to validate correct code installation and performance. The first test case used the Loral Vought developed model data base and corresponding point files. These were run to verify that the hashing code itself worked as a standalone process given known inputs. The second case used the NRC feature extraction code which was hosted on the SSV Sun SPARC computer by a commercial software package which converted the NRC FORTRAN code to C code. This converted code was then manually updated to ensure correct input and output calls and system compatibility. The third check case was a FLIR based run which used the techniques described in check case two, but added the DAP hosted FLIR enhancement algorithms developed by NRC.

5.5 Performance Metrics

The geometric hashing software has been successfully converted and enhanced so that real time system operation is practical . In fact it was successfully demonstrated as part of Demo C.

This new version of the software has a 30-fold decrease in execution time. This time improvement is not due to using faster computational resources since that issue was removed from the calculated improvement factor, but is in fact due to the implementation procedure for the software. The new version is all memory resident and performs no external accesses for any data. The model file is stored in memory and the "live image" point file is supplied to the hashing software from the executive controller. In addition, some other enhancements were made in the way functions were implemented or called to make the code more modular and more efficient. In actuality, the 30-fold decrease in execution time is a conservative number since the present version of the code still has significant intermediate output being presented to the display for the convenience of the developers and system analysts. This intermediate display slows the system significantly and in a final version which would be used in actual field operations as opposed to system technology demonstrations, this intermediate output would be suppressed.

This new version of the software also provides for a 20-fold increase in stored model capability. This increase is not a limit but is the factor which was demonstrated based on the available data to develop the model file. This increase is a direct result of using compact binary files in the actual code operation and having a single model file containing multiple models as opposed to many separate model files.

6. FLIR Hashing Models and Matching Results

For Demo C conducted at Lockheed-Martin, Denver, in July 1995, the NRC/LV team provided a suite of hashing-associated codes for the SSV SPARC processors. The form of that software has been described Section 5. Two separate sensor hash tables were also created: (1) a 72 model LADAR set, which was described in Section 3, and (2) a 66 model FLIR set which will now be addressed.

6.1 FLIR Imagery Used for Model Building

Unlike the LADAR models which were derived from synthetic imagery, the FLIR models were produced from imagery collected by Lockheed Martin on 6 and 7 October 1994 [Munkeby, 1995]. The 3-5 micron Amber FLIR was the same type sensor as that used on the operational SSV's. This FLIR has a square 2.73° x 2.73° narrow FOV resolved into a 256 x 256 pixel array, and thus a resolution of 0.19 milliradians.

The October 6 and 7 data collections were specifically to generate training data for model building, with all targets at a fixed 961 meter range. All targets were on flat, level ground, and the targets were rotated only in azimuth in 30° incremental steps. Each scenario consisted of three targets at a fixed azimuth orientation with one wide FOV image taken of all three targets simultaneously and ten each sequential narrow FOV images for each of the three targets. (For building the FLIR model hash table, we used only one narrow FOV image, chosen randomly, for each target).

The first three collected targets consisted of the M113 APC, the M35 truck, and a HMMWV. Ten scenarios (500-509) were collected on 6 October 1994 and four more scenarios (510-513) on the following day. Table 5 summarizes the image and target truth associated with these scenarios. The height, width, and center values describe the target chip subimage, while the image quality is our subjective assessment of the target's distinctness in the original, unenhanced image. Due to various data collection problems, not all images were actually recorded (as evidenced by no corresponding model number) and the target azimuthal increments do not uniformly change. Such do not represent image deficiencies, but rather only a need for careful book-keeping in cataloging the images and scoring the results.

The second target set was collected entirely on 7 October for ten scenarios (520-529). It consisted of an M543 Wrecker, an M60 Tank, and another HMMWV. Table 6 summarizes the model imagery for this second set, which was collected in a thoroughly consistent order (no missing images, and steadily decreasing azimuth orientations). Unfortunately, the second target set does not include targets at azimuth orientations of 330° and 300°. The data collection had to be terminated prematurely due to rain.

The entire data collection occurred under marginally-acceptable weather conditions (cold, cloudy, and increasingly overcast) with deteriorating weather (impending rain) washing out much of the internal target detail. Such environmental conditions do not correspond with those typical in July, which was when Demo C would occur.

Table 5. First RSTA FLIR Target Training Set - 6 & 7 October 1994

MODEL #	IMAGE	VEHICLE	HEIGHT	WIDTH	X-CENTER	Y-CENTER	ANGLE	QUALITY
0	F500S023	APC	15	19	139	162	270	VG
1	F500S030	TRUCK	20	18	74	162	270	G
2	F500S045	HMMWV	16	18	113	160	270	VG
3	F501S020	APC	17	31	149	162	240	G
4	F501S035	TRUCK	16	30	85	164	240	P
5	F501S042	HMMWV	14	25	110	162	240	P
6	F502S022	APC	18	33	149	156	210	P
7	F502S032	TRUCK	17	38	83	159	210	P
8	F502S043	HMMWV	16	32	114	155	210	VG
9	F503S021	APC	15	32	152	159	180	A
10	F503S038	TRUCK	19	43	85	165	180	A
11	F503S049	HMMWV	18	30	119	162	180	VG
12	F504S023	APC	17	32	160	160	150	VG
13	F504S031	TRUCK	18	42	91	158	150	VG
14	F504S042	HMMWV	17	32	125	160	150	VG
15	F505S021	APC	18	32	176	159	120	A
16	F505S032	TRUCK	19	35	97	165	120	G
17	F505S045	HMMWV	17	25	120	154	120	A
18	F506S022	APC	17	24	171	153	60	A
19	F506S033	TRUCK	21	23	100	160	60	VP
20	F506S040	HMMWV	17	20	128	156	60	P
21	F507S020	APC	19	19	178	166	90	A
22	F507S033	TRUCK	19	19	96	164	90	A
23	F507S043	HMMWV	16	18	130	160	90	A
24	F508S026	APC	17	28	174	164	300	P
	F508S037	TRUCK	19	34	105	165	300	VP
	F508S047	HMMWV	16	28	128	156	300	P
25	F509S028	APC	17	35	174	161	0	A
26	F509S039	TRUCK	18	43	97	157	0	VP
	F509S046	HMMWV	15	33	120	165	0	VP
27	F510S021	APC	14	33	169	164	30	VP
28	F510S032	TRUCK	19	43	75	167	30	VP
29	F510S043	HMMWV	14	32	123	171	30	A
	F511S025	APC	16	34	169	167	0	VP
	F511S031	TRUCK	21	44	72	164	0	VVP
30	F511S040	HMMWV	15	32	129	162	0	A
31	F512S020	APC	17	30	161	160	330	VP
32	F512S038	TRUCK	18	43	66	172	330	VP
33	F512S046	HMMWV	14	32	114	163	330	P
	F513S024	APC	17	28	161	164	300	VVP
34	F513S038	TRUCK	20	32	61	164	300	P
35	F513S048	HMMWV	14	26	122	155	300	G

Originally, we intended to rate the images as either Very Good (VG), Good (G), Average (A), or Poor (P). As we reviewed more of the imagery, we realized that some was of such marginal quality that we needed two additional categories: Very Poor (VP) and Very Very Poor (VVP). Table 7 summarizes this image quality rating for all of the images. Surprisingly, the classification performance (detailed in Section 6.4) was much better overall, as well as for most individual targets (particularly the Wrecker and Tank), than would be inferred from Table 7.

Table 6. Second RSTA FLIR Target Training Set - 7 October 1994

MODEL #	IMAGE	VEHICLE	HEIGHT	WIDTH	X-CENTER	Y-CENTER	ANGLE	QUALITY
36	F520S023	WRECKER	20	19	141	155	270	G
37	F520S038	TANK	20	24	103	164	270	P
38	F520S041	HMMWV	16	18	113	166	270	G
39	F521S020	WRECKER	18	38	141	159	240	A
40	F521S035	TANK	22	38	104	169	240	P
41	F521S047	HMMWV	15	28	116	165	240	A
42	F522S026	WRECKER	20	44	151	165	210	P
43	F522S039	TANK	19	50	97	157	210	VP
44	F522S045	HMMWV	15	32	127	162	210	P
45	F523S023	WRECKER	20	44	151	153	180	P
46	F523S030	TANK	24	49	88	160	180	VVP
47	F523S047	HMMWV	13	30	121	160	180	A
48	F524S021	WRECKER	20	42	159	158	150	A
49	F524S038	TANK	24	47	57	158	150	VP
50	F524S045	HMMWV	13	31	129	155	150	A
51	F525S022	WRECKER	19	35	158	154	120	P
52	F525S036	TANK	18	39	73	158	120	VVP
53	F525S041	HMMWV	15	26	137	166	120	VP
54	F526S025	WRECKER	21	18	168	159	90	VP
55	F526S037	TANK	22	28	69	165	90	VVP
56	F526S046	HMMWV	14	19	134	160	90	VP
57	F527S021	WRECKER	21	38	171	160	60	VP
58	F527S035	TANK	22	40	74	164	60	P
59	F527S042	HMMWV	14	27	135	159	60	VP
60	F528S023	WRECKER	19	49	154	158	30	VP
61	F528S035	TANK	23	50	83	160	30	P
62	F528S042	HMMWV	15	32	139	163	30	VP
63	F529S021	WRECKER	21	52	147	161	0	VP
64	F529S035	TANK	22	61	99	157	0	VP
65	F529S042	HMMWV	15	31	127	165	0	VP

Table 7. Subjective Evaluation of Training Image Quality

	Very Good	Good	Average	Subtotal	Poor	Very Poor	Very Very Poor	Subtotal
APC	14	7	36	57%	14	22	7	43%
TRUCK	7	14	14	35%	22	36	7	65%
HMMWV-1	29	7	28	64%	29	7		36%
WRECKER		10	20	30%	30	40		70%
TANK					40	30	30	100%
HMMWV-2		10	30	40%	10	50		60%
TOTALS	9%	8%	23%	40%	24%	30%	6%	60%

The better than expected performance is probably due in part to the application of image enhancement algorithms, described next, as well as some of the targets having very distinctive shapes for most views.

6.2 Image Enhancement Algorithms

The previous section and particularly Table 7 describes the generally poor image quality of the training imagery that was available for model building. As such, some image enhancement algorithms were applied to the training imagery, as well as being incorporated into the front end of the FLIR hashing software suite installed in the SSV SPARC processors. Figure 9 illustrates these algorithms, with the original image chip shown in the upper left corner. Note the low contrast between this M35 truck and its background.

The first algorithm provides considerable enhancement by simply linearly remapping the original 10 bit imagery to an 8 bit format that is necessary for the other processing algorithms. Next, a standard Histogram Equalization algorithm is applied. Some additional target detail is then achieved by a Spatial Sharpening operator. For that final step, various window sizes (with corresponding weights) compute the average value surrounding the pixel and then, if a threshold is met, subtract the pixel value from that average.

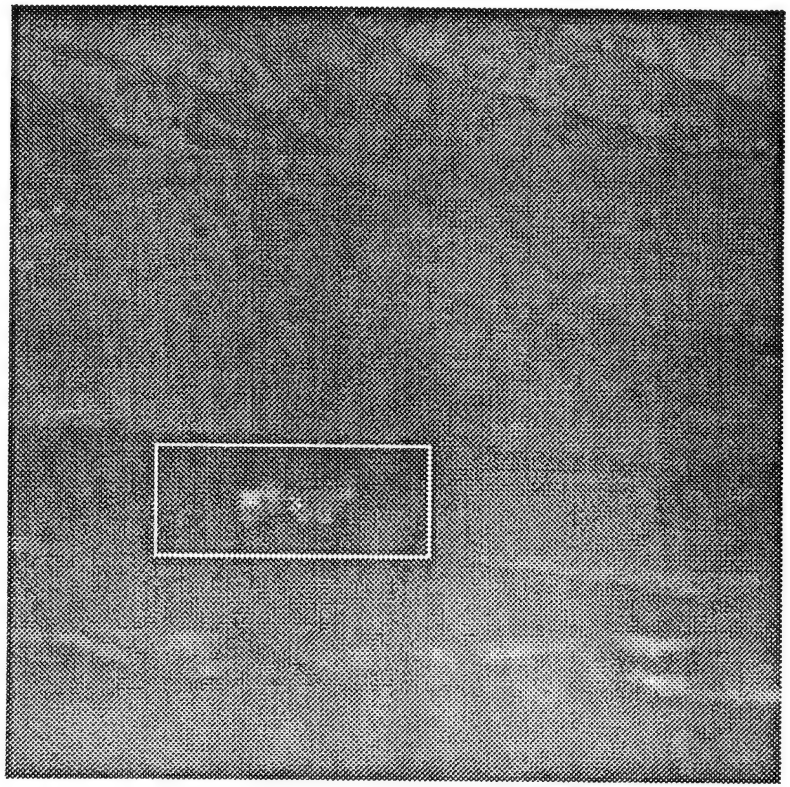
Although it is obvious that this sharpening algorithm produces additional target detail, it is computationally intensive and thus was not included in the operational SSV algorithm suite to process unknown, live imagery. Rather, the sharpening is only used in the off-line, model building process where increased computational time is not deleterious.

The bottom four subimages shown in Figure 9 are not part of the Image Enhancement suite, but rather correspond to the subsequent processing steps, as already illustrated (for LADAR imagery) in Figures 4-6. These additional subimages are included to show how well a hash point set can be generated (from the enhanced image) which provides a good geometric representation of the truck.

Also note that the final set of model hash points is not the same as those produced by the point extraction algorithm. For the model building only (as opposed to the processing of the live image), the human analyst is given the opportunity to add and/or delete points from the original extracted set. This is typically done so that the model provides the best possible geometric representation that the radiometric conditions would allow. Hence, points missing due to occlusion, poor contrast, or lack of line curvature can be added to the model set. Conversely, occasional extraneous points from the immediate background that are initially extracted can be deleted.

6.3 Hash Point Models

Using the process illustrated in Figure 9 (in the previous section), hash point models were created for the 66 FLIR image scenarios collected on 6 and 7 October. The corresponding images are delineated in Tables 5 and 6 in Section 6.1. The following five figures, by target type,



Original FLIR Image with Detection Window

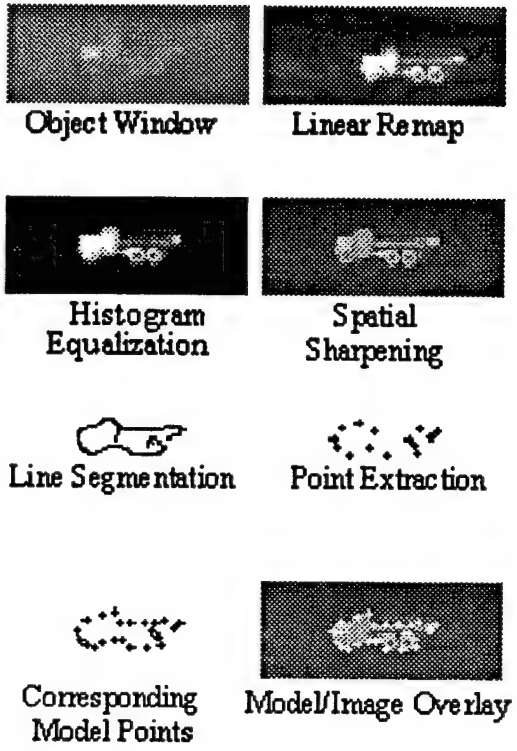


Figure 9. FLIR Image Enhancement and Geometric Hashing of an M35 Truck

show the target at each of its orientations and the extracted hash point model that was created. Note that in Figures 10 through 14, the target orientations are presented in the order in which they were collected, which is not always in a monotonic angle order.

Figure 10 provides two views of the HMMWV for most of its orientations, since imagery was collected for that target as two different sets. The first set provides all twelve 30° azimuth steps, whereas only ten were collected in the second set; as noted in Section 5.1, imagery at the 330° and 300° orientations was not collected during the second set.

Figures 11 and 12 provide the twelve views and corresponding point models for the two other targets collected during the first set: the M113 APC and the M35 truck. Figures 13 and 14 provide the ten views (again the 330° and 300° orientations are absent) for the other two targets in the second set: the M543 Wrecker and the M60 tank.

Inspection of Figures 10-14 shows that in many instances the point models do not exactly mimic the target geometry. This is due mainly to the lack of sufficient quality in many of the images. Also, the Line and Point Extractors are not perfect, even when the image quality is very good. (In this respect some modest improvements to those extractors has since been initiated). Notwithstanding these degraded point representations, they are nonetheless sufficient in almost all instances to provide a unique representation by target type and orientation. Hence, it should not be too surprising that excellent classification results were obtained against this 66 model set, as discussed next.

6.4 FLIR Hashing Test Results

The 66 FLIR model hash table was initially tested at Demo C, for which two of the three "unknown" targets (M113 APC, HMMWV, and M35 Truck) were correctly recognized. The very limited Demo C schedule did not allow additional target types to be tested. The key match criteria were:

100 percent model points used
100 percent live points used
50 percent live points matched
1 pixel mismatch tolerance
1.4 average pixel mismatch
10° maximum in-plane rotational angle disparity

The explanation of these parameters is given in [Akerman, 1994].

To quantify more thoroughly the FLIR hashing performance, an extensive laboratory experiment was subsequently conducted using the same 66 model hash table. The original images from which those models were derived were input into the overall hashing suite (except the second HMMWV target was not input). Hence, there were 56 trials, for each of which there was a complete process of image enhancement (but without spatial sharpening), line extraction, and

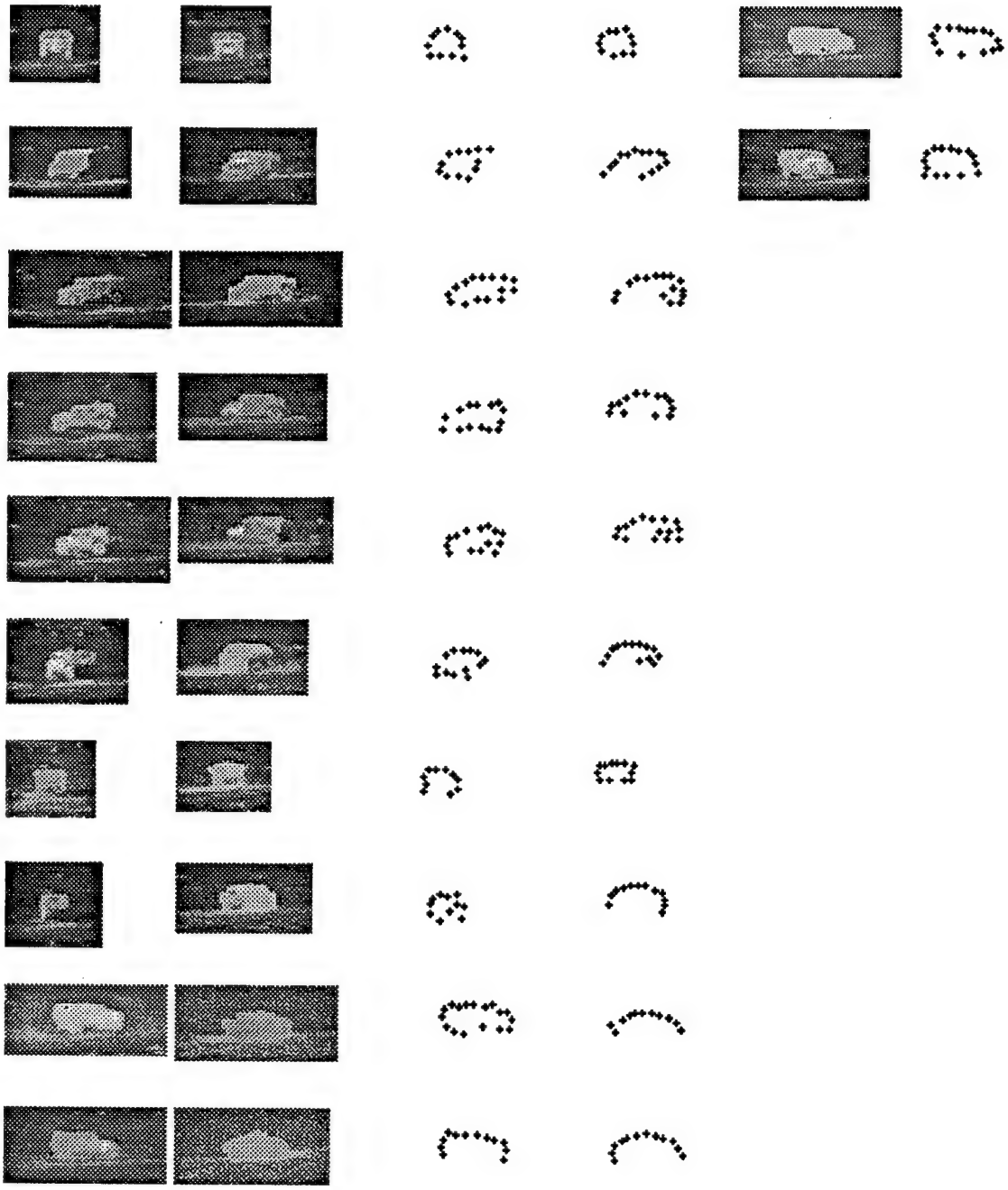


Figure 10. HMMWV Images and Point Models

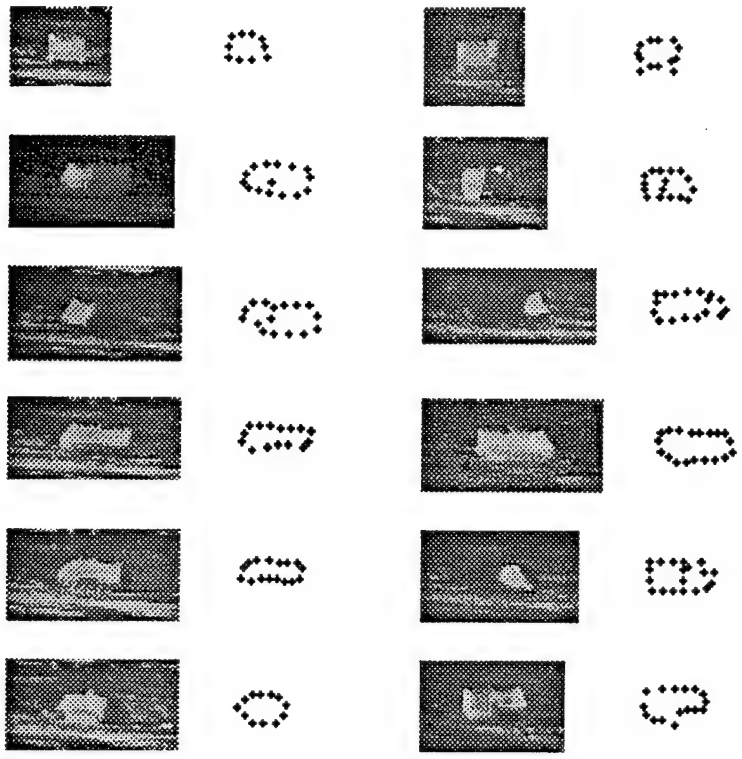


Figure 11. M113 APC Images and Point Models

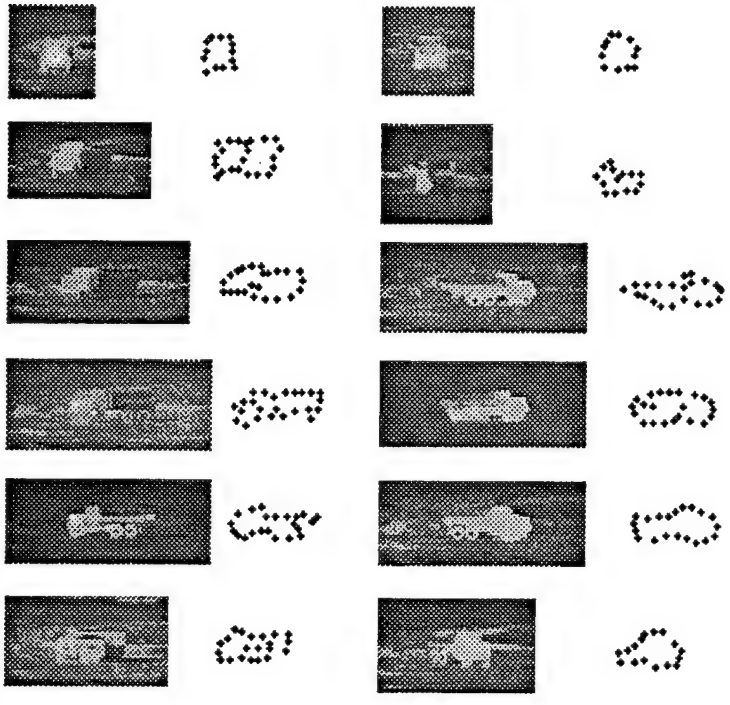


Figure 12. M35 Truck Images and Point Models

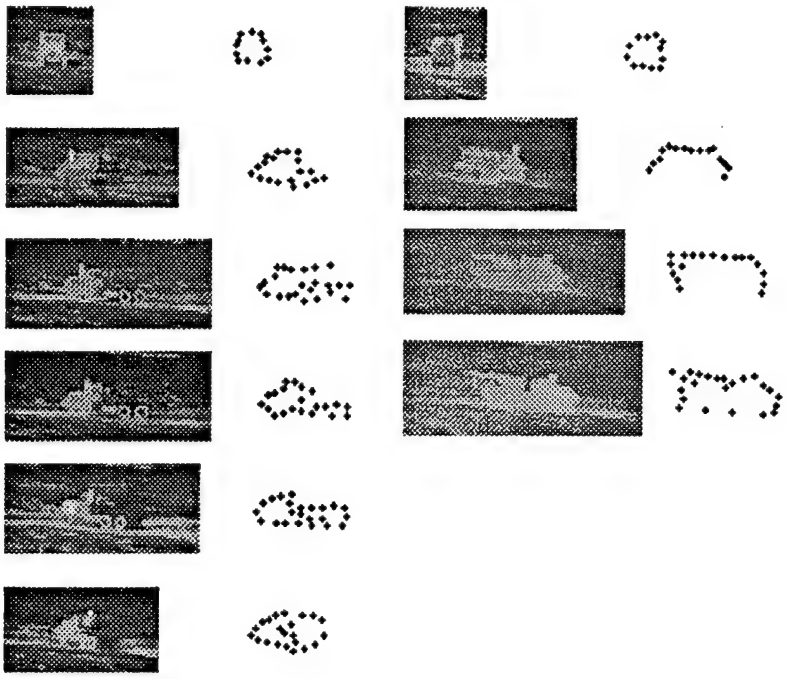


Figure 13. M543 Wrecker Images and Point Models

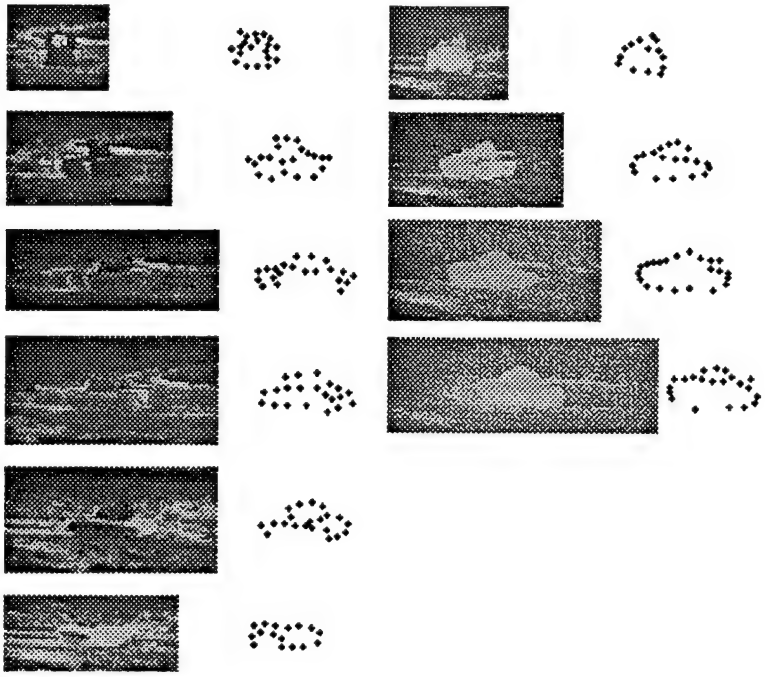


Figure 14. M60 Tank Images and Point Models

point extraction. Those automatically extracted points (without any human adjustment) were then tested against the 66 model hash table. The same matching parameters as those of Demo C were used, except the percent of live points to be matched was reduced to 20%.

Tables 8 and 9 give the resultant classification matrices, both for absolute numbers and corresponding classification probabilities. **The overall average classification probability is 86%, which includes all trucks and wreckers being correctly classified.** These two tables include only the live HMMWV input images from the first set of training data (6 October 94) since the second set is a redundant target, as well as being of very poor image quality. Referring back to Table 3, one will note that the HMMWV image quality is rated better than poor as 64% on 6 October, but only 40% on 7 October. These percents correspond almost identically with the classification results for that target for each day.

Tables 10 through 14 give the results of each trial, with each table corresponding to one of the five target types. The table headings correspond to the match criteria, which are thoroughly discussed in [Akerman, 1994]. However, the following elaboration may be helpful:

- Target Azimuth - This is the orientation of the target in the live (unknown) image. Due to the sequence of data collection, those angles are not in the same order in each table. If the angle is in ( )'s, this signifies that this target was not properly classified.

- Live, Model, and Match Points - The primary decision criteria is maximum percent of live points matched. Some classification errors may have been avoided if the criteria for percent of model points matched had also been imposed.

- Average (Pixel Mismatch) Distance and (Maximum In-Plane Rotation) Angle Difference - Note that when these values get close to the maximum permitted, 1.4 pixels and 10 degrees, then the outcome is more likely to be a misclassification. Such is particularly true when both numbers are close to the threshold limits.

- Model and Live Basis Distances - These are the pixel distances between the master and slave points in the Live image and the Model to which it is matched. When these numbers are not the same, and particularly when they differ by a factor of two or three, this again is an indicator of a misclassification.

None of the above criteria are sufficient just in themselves to improve the classification performance over that achieved by the existing criteria. However, an examination of the data in Tables 10-14 suggests that a more intelligent selection of criteria and their threshold values could further improve performance.

Table 8. Overall Classification Matrix

	APC	Truck	Tank	Wrecker	HMMWV
APC (M113)	9	3	----	----	----
Truck (M35)	----	12	----	----	----
Tank (M60)	----	1	9	----	----
Wrecker (M543)	----	----	----	10	----
HMMWV	1	----	1	2	8

Table 9. Overall Classification Probability Matrix (P_{cc} = 86%)

	APC	Truck	Tank	Wrecker	HMMWV
APC (M113)	0.75	0.25	0	0	0
Truck (M35)	0	1.00	0	0	0
Tank (M60)	0	0.10	0.90	0	0
Wrecker (M543)	0	0	0	1.00	0
HMMWV	0.08	0	0.08	0.17	0.67

Table 10. Classification Results for M35 Truck Target (12 of 12)

	Target	---------Points---------			Average	Angle	Basis Distance		Classification	
Image #	Azimuth	Live	Model	Match	Distance	Difference	Model	Live	Type	Model #
1	270	9	13	8	0.0	0	12	12	Truck	1
2	240	10	22	10	0.33	1	20	18	Truck	4
3	210	10	23	10	0.0	0	9	9	Truck	7
4	180	15	24	13	0.0	0	3	3	Truck	10
5	150	14	22	14	0.34	1	22	28	Truck	13
6	120	13	21	11	0.14	0	3	3	Truck	16
7	60	10	14	8	0.63	5	8	4	Truck	19
8	90	8	12	8	0.0	0	13	13	Truck	22
9	(0)	16	22	16	0.0	0	3	3	Truck	26
10	30	15	21	11	0.0	0	11	11	Truck	28
11	330	15	20	15	0.0	0	36	36	Truck	32
12	300	15	16	12	0.0	0	24	24	Truck	34

Table 11. Classification Results for HMMWV Target (8 of 12)

[36% of images at least poor]

Image #	Target Azimuth	Points			Average Distance	Angle Difference	Basis Distance		Classification	
		Live	Model	Match			Model	Live	Type	Model #
1	270	6	11	6	0.0	0	12	12	H	2
2	(240)	10	20	9	1.03	2	14	19	Tank	37
3	(210)	9	19	7	0.5	9	25	9	W	39
4	180	11	14	10	0.0	0	22	22	H	11
5	150	10	14	8	0.0	0	11	11	H	14
6	(120)	8	19	8	0.83	4	21	8	APC	31
7	(60)	8	23	8	0.98	8	15	5	W	48
8	90	5	10	5	0.0	0	12	12	H	23
9	30	12	17	12	0.0	0	11	11	H	29
10	0	7	12	7	0.0	0	5	5	H	30
11	330	9	15	9	0.0	0	20	20	H	33
12	300	11	15	10	0.0	0	22	22	H	35

Table 12. Classification Results fro M113 APC Target (9 of 12)

Image #	Target Azimuth	Points			Average Distance	Angle Difference	Basis Distance		Classification	
		Live	Model	Match			Model	Live	Type	Model #
1	270	6	10	6	0.0	0	13	13	APC	0
2	240	10	15	10	0.0	0	4	4	APC	3
3	(210)	10	20	7	0.88	3	3	17	Tank	37
4	180	10	15	10	0.0	0	26	26	APC	9
5	150	8	16	8	1.12	2	8	17	APC	12
6	(120)	7	20	6	0.6	9	9	27	Tank	37
7	60	6	16	6	0.0	0	5	5	APC	18
8	90	9	13	9	0.0	0	10	10	APC	21
9	300	11	16	11	0.0	0	23	23	APC	24
10	0	10	19	10	0.0	0	7	7	APC	25
11	30	6	17	6	0.0	0	8	8	APC	27
12	(330)	5	20	5	1.31	4	14	15	Tank	37

Table 13. Classification Results for M543 Wrecker Target (10 of 10)

Image #	Target Azimuth	Points			Average Distance	Angle Difference	Basis Distance		Classification	
		Live	Model	Match			Model	Live	Type	Model #
1	270	6	12	6	0.0	0	9	9	W	36
2	240	11	19	9	0.0	0	14	14	W	39
3	210	15	23	13	0.08	0	7	7	W	42
4	180	10	21	9	0.0	0	10	10	W	45
5	150	9	23	8	0.0	0	27	27	W	48
6	120	11	22	11	0.0	0	5	5	W	51
7	90	10	12	9	0.0	0	16	16	W	54
8	60	11	15	9	0.0	0	23	23	W	57
9	30	13	18	7	0.24	0	15	18	W	60
10	0	18	23	14	0.0	0	10	10	W	63

Table 14. Classification Results for M60 Tank Target (9 of 10)

Image #	Target Azimuth	Points			Average Distance	Angle Difference	Basis Distance		Classification	
		Live	Model	Match			Model	Live	Type	Model #
1	270	10	20	10	0.87	2	8	5	Tank	37
2	240	15	21	11	0.0	0	25	25	Tank	40
3	210	13	21	8	0.63	1	25	43	Tank	43
4	(180)	11	22	9	1.08	9	11	21	Truck	13
5	150	16	21	12	0.09	0	11	11	Tank	49
6	120	9	16	8	0.0	0	25	25	Tank	52
7	90	11	14	10	0.0	0	13	13	Tank	55
8	60	11	19	11	0.0	0	28	28	Tank	58
9	30	20	22	16	0.0	0	20	20	Tank	61
10	0	13	21	11	0.0	0	18	18	Tank	64

7. List of Publications and Technical Reports

1. A. Akerman III, "MultiSensor Fusion Using FLIR & Ladar Identification," NRC-94-052, 12 April 1994.

2. A. Akerman III, R. Patton, W. Delashmit, and R. Hummel, "MultiSensor Fusion Using FLIR and Ladar Identification", Image Understanding Workshop, Monterey, CA, 15 November 1994.

3. A. Akerman III, R. Patton, W. Delashmit, and R. Hummel, "MultiSensor Fusion Using FLIR and Ladar Identification," 8th National Sensor Fusion Symposia, Dallas, TX, 17 March 1995.

4. A. Akerman III, R. Patton, W. Delashmit, and R. Hummel, "Target Identification Using Geometric Hashing and FLIR/Ladar Fusion," Image Understanding Workshop, Palm Springs, CA, February 1996.

5. A. Akerman III, R. Patton, W. Delashmit, R. Hummel, T. Strat, and T. Witten, "Geometric Hashing for Three Types of Sensor Imagery," 5th ATR Systems and Technology Symposia, John Hopkins University Applied Physics Laboratory, 23-25 July 1996.

6. A. Akerman III, R. Patton, W. Delashmit, and R. Hummel, "Target Identification Using Geometric Hashing and FLIR/LADAR Fusion," in Chapter 4, Target Detection and Recognition, in Reconnaissance, Surveillance, and Target Acquisition for the Unmanned Ground Vehicle, Dr. Oscar Firschein, ed., in press.

8. List of Presentations and Demonstrations

1. A. Akerman III, W. Delashmit, and R. Hummel, "Reconnaissance, Surveillance, and Target Acquisition" ARO program kickoff meeting, Durham, NC, 6-8 October 1993.

2. A. Akerman III, "ARPA/ARO Technical Exchange and Image Metrics Meeting," Fairfax, VA 13-15 March 1994.

3. A. Akerman III, R. Patton, W. Delashmit, and R. Hummel, "Multisensor Fusion Using FLIR and Ladar Identification," poster board, video tape, and computer presentation at SSV Demonstration B, Lockheed Martin, Denver, CO, 8 June 1994.

4. A. Akerman III, R. Patton, and W. Delashmit, "Multisensor Fusion Using FLIR and Ladar Identification," operational vehicle demonstrations and poster board presentation at SSV Demonstration C, Lockheed Martin, Denver, CO, 25-28 July 1995.

5. W. Delashmit, "Multisensor Fusion Using FLIR and Ladar Identification," poster board, video tape, and computer presentation at DARPA/SISTO Symposia, Fairfax, VA, 28-31 August 1995.

6. R. Patton and W. Delashmit, "Reconnaissance, Surveillance, and Target Acquisition," poster board and computer demonstration at SSV Demonstration II, Ft. Hood, TX, June 19-20, 1996.

7. A. Akerman III, R. Patton, W. Delashmit, R. Hummel, and H.R. Myler, "Classification of FLIR and LADAR Imagery Using Geometric Hashing and Recognition by Components," DARPA Image Understanding Technology Program Reviews, USA CECOM NVESD, Ft. Belvoir, VA, 16-17 September 1996.

9. List of Participating Scientific Personnel

a. Nichols Research Corporation
   (1) Alexander Akerman III (Principal Investigator)
   (2) Ronald Patton (Consultant)
   (3) Dr. Harley R. Myler (Consultant)

b. Lockheed Martin Vought Systems
   (1) Walter Delashmit
   (2) Bryan Pettitt
   (3) Phillip Thompson

c. New York University
   (1) Dr. Robert A. Hummel

10. References

Akerman, A. Patton, R., Delashmit, W., and Hummel, R., "Multisensor Fusion Using FLIR and LADAR Identification," 1994 RSTA Technical Reports of the ARPA Image Understanding Program, 101-123; also in 8th National Sensor Fusion Symposia, March 1995.

Akerman, A., Patton, R., Smith, S., and Williams, A., FLIR and MMW Algorithms to Detect and Classify Stationary Targets, I-MATH Associates, Inc. Report 93IR36, Contract DAAB07-93-U-0006, USA NVESD, Ft. Belvoir, VA, 8 November 1993; also in Proceedings of the 4th ATR Systems and Technology Symposia, ATRWG PR-94-001, Vol. I, pp. 61-90, November 1994.

Akerman, A., "Pyramidal Techniques for Multisensor Fusion," SPIE Proceedings 1828-15, O/E Tech 92 Symposia, November 1992.

Akerman, A., Patton, R., and Mayes, W., Parallel Processing Application for Avionics, USAF-WL-TR-92-1063, May 1992.

Akerman, A. and Patton, R. "Model-Based Vision Using Geometric Hashing," 1990 Applied Imagery and Pattern Recognition Workshop, SPIE Proceedings 1406-07, November 1990.

Biederman, I., "Recognition-By-Components: A Theory of Human Image Understanding," Psychological Review, Vol. 94, No. 2, pp. 115-147, 1987.

Bourdon, O. and Medioni, "Object Recognition Using Geometric Hashing on the Connection Machine," Proceedings of the ICPR, 1990.

Burt, P.J. and Adelson, E.H., "The Laplacian Pyramid as a Compact Image Code," IEEE Trans COM, 31, (4), April 1983.

Lamdan, Y. And Wolfson, J.J., "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," Proceedings of the Second International Conference on Computer Vision, pp. 238-249, December 1988.

Lowe, D.G., Perceptual Organization and Visual Recognition, Kluwer Academic Publishers, ISNB 0-89838-172-X, 1985.

Munkeby, S.H., RSTA September 94 Data Collection, SSV Report, ARPA Order 8353, Martin Marietta Astronautics, January 31, 1995.

Severson, W., "RSTA Interface Files," Martin Marietta Technologies, Inc., 27 February 1995.

Thomopoulos, et al, "Optimal Decision Fusion in Multiple Sensor Systems," IEEE Trans AES-23, No. 5, pp. 644-653, September 1987.

Toet, A., Van Ruyven, L.J., and Valeton, J.M., "Merging Thermal and Visual Images by a Contrast Pyramid," Optical Engineering, 28, (7), July 1989.

Appendix A. Geometric Hashing Evolution

The original idea of geometric hashing comes from the research work of matching boundary curves [Kalvin et al., 1986]. The research done by Schwartz and Sharir [1987], Wolfson [1987], Hong and Wolfson [1988], all rely on the technique of geometric hashing. They develop the technique of finding invariants for boundary curves that are called footprints.

Lamdan and Wolfson [1988] give a description of the geometric hashing method. Early prototype systems for recognizing flat industrial parts and synthesized 3D objects are reported by Lamdan et al. [1988a, 1988b, 1990]. The features are called "interest points." That is, the geometric hashing method performs point pattern matching in these experiments. Gavrila and Groen [1992] use a geometric hashing system to recognize 3D CAD models.

A parallel implementation of geometric hashing on the Connection Machine is reported by Rigoutsos and Hummel [1991a, 1992], and also one by Khokhar and Prasanna [1993]. Rigoutsos and Hummel [1993] also report a distributed version of geometric hashing for object recognition.

Rigoutsos and Hummel [1991b, 1991c] assume the appearance of Gaussian noise for the position of the point pattern and derive analytic solutions for the features in hash space. A precise weighted voting formula with a Bayesian interpretation for geometric hashing is given. Tsai [1993] analyzes the affine invariants for line features. Line features are represented as a point in (θ,r) space.

Grimson and Huttenlocher [1990] analyze the performance of geometric hashing by assuming that the noise model of the feature points is an ε-disc. Lamdan and Wolfson derive the false alarm rate empirically and analytically. Their analysis is performed on (r,θ) space with bounded error model. Sarachik [1992] and Sarachik and Grimson [1993] investigate the performance of geometric hashing with the assumption of a Gaussian noise model. They obtain predictions of operating characteristics of simple recognition systems, which show acceptable performance under low-noise conditions.

Califano and Mohan [1991, 1994] use higher-order features to improve the performance as well as the fault tolerance of the recognition system. Liu and Hummel [1994] also adopt the strategy of using higher order features. The features are attributed with extra information. The discrimination power of using attributed features are improved so that a 3D object embedded in a complicated background can still be recognized.

Bibliography

Califano, A. and Mohan, R., "Multidimensional Indexing for Recognizing Visual Shapes," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 1991.

Califano, A., and Mohan, R., "Multidimensional Indexing for Recognizing Visual Shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(4), pp. 373-392, April 1994.

Gavrila, D.M. and Groen, C.A., "3D Object Recognition From 2D Images Using Geometric Hashing," Pattern Recognition Letters, 13(4), pp. 263-278, April 1992.

Grimson, W. and Huttenlocher, D., "On the Sensitivity of Geometric Hashing," International Conference on Computer Vision, pp. 334-338, Osaka, Japan, December 1990.

Hong, J. and Wolfson, H.J., "An Improved Model-Based Method Using Footprints," Proceedings of the 9th International Conference on Pattern Recognition, pp. 72-78, 1988.

Khokhar, A. and Prasanna, V.K., "Scalable Data Parallel Geometric Hashing: Experiments on MasPar MP-1 and on Connection Machine CM-5," Proceedings of the DARPA Image Understanding Workshop, pp. 851-859, 1993.

Kalvin, A., Schonberg, E., Schwartz, J.T., and Sharir, M., "Two-Dimensional, Model-Based, Boundary Matching Using Footprints," The International Journal of Robotics Research, 5(4), pp. 38-55, 1986.

Liu, J.J. and Hummel, R., "Geometric Hashing with Attributed Features," Proceedings of the Second CAD-Based Vision Workshop, Champion, Pennsylvania, February 1994.

Lamdan, Y., Schwartz, J.T., and Wolfson, H.J., "Object Recognition By Affine Invariant Matching," Proceedings Computer Vision and Pattern Recognition, pp. 335-344, 1988.

Lamdan, Y., Schwartz, J.T., and Wolfson, H.J., "On Recognition of 3-D Objects From 2-D Images," Proceedings IEEE International Conference on Robotics and Automation, pp. 1407-1413, 1988.

Lamdan, Y., Schwartz, J.T., and Wolfson, H.J., "Affine Invariant Model-Based Object Recognition," IEEE Transactions on Robotics and Automation, 5(6), pp. 578-589, October 1990.

Lamdan, Y. and Wolfson, H.J., "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," Proceedings of Second International Conference on Computer Vision, pp. 238-249, 1988.

Rigoutsos, I. and Hummel, R., "Implementation of Geometric Hashing on the Connection Machine," IEEE Workshop on Directions in Automated CAD-Based Vision, Maui, Hawaii, June 1991.

Rigoutsos, I. and Hummel, R., "Robust Similarity Invariant Matching in the Presence of Noise," Proceedings of the 8th Israeli Conference on Artificial Intelligence and Computer Vision, Tel Aviv, Israel, December 1991.

Rigoutsos, I. and Hummel, R., "Several Results on Affine Invariant Geometric Hashing," Proceedings of the 8th Israeli Conference on Artificial Intelligence and Computer Vision, Tel Aviv, Israel, December 1991.

Rigoutsos, I. and Hummel, R., "Massively Parallel Model Matching: Geometric Hashing on the Connection Machine," IEEE Computer: Special Issue on Parallel Processing for Computer Vision and Image Understanding, February 1992.

Rigoutsos, I. and Hummel, R., "Distributed Bayesian Object Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York City, New York, June 1993.

Sarachik, K.B., "Limitations of Geometric Hashing in the Presence of Gaussian Noise," Technical Report 1395, M.I.T. AI Lab, October 1992.

Sarachik, K.B. and Grimson, W.E.L., "Gaussian Error Models for Object Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York City, New York, June 1993.

Schwartz, J.T. and Sharir, M., "Identification of Partially Obscured Objects in Two and Three Dimensions By Matching Noisy Characteristics Curves," The International Journal of Robotics Research, 6(2), pp. 29-44, 1987.

Tsai, F.C.D., "Robust Affine Invariant Matching with Application To Line Features," Computer Vision and Pattern Recognition, pp. 393-399, IEEE Computer Society, June 1993.

Wolfson, H., "On Curve Matching," Proceedings of IEEE Workshop on Computer Vision, pp. 307-310, 1987.

Appendix B. Theoretical Formulation for Hashing of Ladar Imagery

B.1 Model Building with Depth Values and Corner Points

The hash table is constructed that encodes the information about the models in a view-centered fashion. Especially because we are dealing with 3D information, it may be possible to use a different model representation strategy. However, our first object recognition strategy uses separate models for every viewing direction. Accordingly, we begin separate models for each target type, for each discretized viewing direction. The viewpoint direction of the model is a two-parameter collection of locations on the "viewing sphere," although in our initial experiments, we will assume a constant depression angle, and thus the viewpoint direction reduces to a single parameter.

The data that are encoded for each model are of two types: relative depth data and corner discontinuities. That is, for each model, we form two sets of data, using predictions based on the model. One set consists of the depth information at a finely-quantized two dimensional grid of points, resulting in a set $\{(x_i, y_i, z_i)\}$ of depth values. The location of the origin for this collection is unimportant, since the values will only be used in terms of differences. The second set of data consists of locations of corners that are predicted to be visible along depth discontinuities, and can be represented as a collection of two-dimensional locations $\{(x_i,y_i)\}$. The corner data can optionally be attributed with extra information, such as a predicted orientation of the angle bisector of the corner, when projected onto the image plane. In this case, the data takes the form $\{(x_i,y_i,\theta_i)\}$. We reiterate that this information is dependent on the model m, and that a model is a target/orientation pair.

Next, we choose basis sets. A single (x,y,z) location suffices to determine a basis set. Theoretically, we could use all of the depth data as potential basis points, but we instead will limit the size of the hash table and the number of representations of the model by choosing only 3D locations corresponding to corner detections. That is, for every predicted corner location $(\bar{x}_i, \bar{y}_i)$, we find a corresponding (x_{ji}, y_{ji}, z_{ji}) in the depth data that has the same (or nearly the same) (x,y) coordinates, and we consider the index i as a possible basis index for the model m. The actual basis for index i is located at (x_{ji}, y_{ji}, z_{ji}).

We then form hash table entries for the model/basis pair (m,j_i). There are essentially two hash tables, corresponding to the two kinds of data. The depth hash table consists of entries

$$\omega_k(m,i) = (x_k, y_k, z_k) - (x_{ji}, y_{ji}, z_{ji})$$

for all $k \neq j_i$. That is, each position in the model is measured relative to the 3D location of the basis point, and the resulting normalized positions become hash table entries for the particular model with the particular basis.

For the corner data, we construct entries from the predicted observable corners in the Ladar data, normalizing with respect to the (x,y) locations. Thus for every (x_k, y_k, θ_k) encoding a corner location in the model m, we form a hash entry

$$\overline{\omega_k}(m,i) = (\overline{x_k} - x_{ji}, \overline{y_k} - y_{ji}, \theta_k)$$

Thus the corner data entries are relative (x,y) positions with respect to the basis point location, together with the predicted angular bisector direction of the corner.

The entries should additionally be endowed with covariance information; i.e., predictions about the variations of the hash values due to inaccuracies in sensing. This information is needed in order to ensure that the weighted voting geometric hashing scheme properly implements a Bayesian reasoning system, under the assumption that the hash values of the observed scene data provide independent information (a conditional independence assumption). For our preliminary studies, we will use a simplified covariance estimation procedure. Namely, for the hash table entries $\omega_k(m,i)$, we assume a spherical distribution of values centered at the 3D location of the entry, with standard deviation proportional to the Euclidean norm of the hash value entry. For the corner data, the entry $\omega_k(m,i)$ is assumed to have circular variation in the (x,y) components with standard deviation proportional to (but with a larger constant of proportionality) the Euclidean distance from the origin, and the θ component is presumed to be statistically independent and Gaussian distributed with a fixed variance.

B.2 New Voting Schema

Data is obtained on a far coarser sampling rate, and with much greater noise than in the case of the model data. Nonetheless, we are able to extract lines, corners, and have readily available depth values from the observed objects.

We use a corner detector to obtain potential basis points. Currently, we are using the C++ version of the Cox-Boie edge detector, and the line following and coalescing routines. We have ported the Cox-Boie edge detector to KHOROS, displaying the results with Cantana.

In any case, image locations where corners are detected are located. We pick one such point as a candidate basis location (at location, say, (x_0, y_0, z_0)), and we perform a trial. The algorithm must iterate over trials until all interesting locations have been explored. In a trial, we perform hashing of the detected object subimage and weighted voting of the model/basis candidates. Hashing works as follows.

For all pixel locations (x,y,z) near the basis point , (x_0, y_0, z_0) in the scene, we compute a relative $(\xi, \eta, \zeta) = (x,y,z) - (x_0, y_0, z_0)$ value for each such point. The coordinate values correspond to a differential distance from the observed basis point location in the scene. When computing the depth value z_0 for the basis point, we use a local minimum of range values in order to be sure that the range is obtained for the foreground object, and not the background. Each such (ξ, η, ζ) location becomes a hash value that hashes into the three-dimensional range data hash table. We need only concern ourselves with (ξ, η, ζ) values that are sufficiently small that they could plausibly be on the same target as the basis point.

Likewise, nearby extracted corners are used to compute a location (ξ,η, θ) giving a relative position to the basis point and the orientation of the angle bisector. This value hashes into the three-dimensional corner-values hash table.

For each range-based hash, say (ξ, η, ζ), nearby entries are located in the hash table. For each entry of the form $\omega_\eta(m,i)$ that is located near (ξ, η, ζ) a search is made for the entry $\omega_k(m,i)$ that is closest to (ξ, η, ζ). Since the entries of the form $\omega_q(m,i)$ form a "sheet"

representing the surface of the object, they will be located quite densely, and the entry $\omega_k(m,i)$ that is nearest (ξ, η, ζ) will be the nearest point on this surface.

Recall that $\omega_k(m,i)$ is located at (x_k, y_k, z_k) - (x_{ji}, y_{ji}, z_{ji}). This entry then receives a vote, which replaces its current vote only if it is greater than its current vote. All votes are initially zero. The vote for entry $\omega_k(m,i)$ is denoted by $z_k(m,i)$, and the vote amount, for the depth data, depends on the distance from the point (ξ, η, ζ) to the sheet, at point $\omega_k(m,i)$. If the (x,y) coordinate locations are far apart, then the observed point is not occurring "in front" of the model, and the vote will be zero. However, ordinarily, if there is one point of the sheet nearby, then the nearest point will be perpendicular to the hash point, which in the nearly orthogonal projection, means that the (x,y) components nearly match. In this case, the distance d is essentially the different in the z components.

The vote should be large if this distance d is small, and will be negative if the distance is large. The Bayesian theory says that the value should be

$$z_k(m,i) = \log\left[\frac{Prob((\xi,\eta,\zeta)|(m,i),(x_0,y_0,z_0))}{Prob((\xi,\eta,\zeta))}\right]$$

where the Prob's measure density distribution values at the location of the hash, and the condition in the numerator means that it is known that the model m appears with basis point i at location (x_0, y_0, z_0). To model this vote, we use the formula

$$z_k(m,i) = \log\left(\frac{\frac{1}{\sqrt{2\pi\sigma_1}}e^{-d^2/2\sigma_1^2}}{\frac{1}{\sqrt{2\pi\sigma_2}}e^{-d^2/2\sigma_2}}\right) = c_1 - c_2 d^2$$

where d is the distance between (ξ, η, ζ) and $\omega_k(m,i)$ and σ_1 and σ_2 are constants discussed below.

The value σ_1 is expected depth variation (the standard deviation value, actually) due to sensor noise, measurement noise, and also changes in the vehicle at any given location. The units are in length and so for a high quality sensor, are likely to be on the order of a foot or two. The value of σ_2 is the standard deviation for point to point variations of depth, without any other knowledge. The value of C_1 is $(1/2)\log(\sigma_2/\sigma_1)$, and the coefficient C_2 is simply $(1/2\sigma^2_2)$-$(1/2\sigma^2_1)$. Presumably, the weighted vote should saturate at some negative amount, and not get too negative, reflecting the fact that a sensor drop-out is possible. Also, this formula could easily be modified to account for the fact that the σ_1 value should be larger for positive values of d, (representing the possibility of occlusion of the model) than for negative values of d (which would occur when the model has a hole in it).

For hashes of corner detections, a similar formula operates. That is, a hash to location (x,y,θ) is used to locate nearby entries of the form $\bar{\omega}_k(m,i)$. In this case, because corner detections are well separated for any given model/basis combination, there is no need to search for the nearest $\bar{\omega}$ entry with model/basis (m,i). A weighted vote $\bar{z}_k(m,i)$ is recorded for

the entry. This time, the "distance" between the hash point and the entry can be measured by a weighted sum of the square distance in the (x,y) plane, and the square difference in the θ variable. The z component plays no role because it has already been accounted in the depth hashes. The weights will depend on the expected variations. Let d^2 represent the weighted sum of square differences. That is,

$$d^2 = a_1\left[(\overline{x_k} - x_{ji} - x)^2 + (\overline{y_k} - y_{ji} - y)^2\right] + a_2(\theta_k - \theta)^2$$

Here, the weights a_1 and a_2 will have to be determined empirically. Then the formula for the weighted vote is similar to before:

$$\overline{z_i}(m, j_i) = \overline{c_1} - \overline{c_2}d^2$$

Again, the value should be clipped if it becomes too negative. Also, only corners near the basis point need be considered. Here, the $\overline{C}_1$ and $\overline{C}_2$ values depend on two standard deviation values, σ_1 and σ_2, just as above, where the first represents expected distances of the corners from nearby corner entries given knowledge of the placement of the model, and the σ_2 entry corresponds to a priori distance deviations.

Finally, votes are combined. The total weighted vote for any given model/basis is a sum of the weighted votes for all entries bases on the model/basis:

$$W(m, j_i) = \sum_i z_i(m, j_i) + \sum_i \overline{z_i}(m, i)$$

This sum is performed over all model/basis sets, and model/bases that receive a large weighted vote are candidate detections.

The result is that a model that is likely to be present will receive a large corresponding vote for some (m,i) pair, providing the chosen basis location , (x_0, y_0, z_0) lies near a corner of a model point. We thus see that it is extremely important to be able to extract from the detected subimage basis points (in our case, corner points) that correspond to corner points pre-stored as basis points in the models.